SURVEILLANCE VERSUS RECONNAISSANCE:
AN ENTROPY BASED MODEL

THESIS

Tamilyn S. Dismukes, Major, USAF

AFIT/OR-MS/ENS/12-09

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/OR-MS/ENS/12-09

SURVEILLANCE VERSUS RECONNAISSANCE: AN ENTROPY BASED MODEL

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Operations Research

Tamilyn S. Dismukes, BS

Major, USAF

March 2012

Distribution Statement A
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

AFIT/OR-MS/ENS/12-09

SEARCH VERSUS RECONNAISSANCE:
AN ENTROPY BASED MODEL

Tamilyn S. Dismukes, BS
Major, USAF

Approved:

________// signed //____________________	__1 Mar 12__
Darryl K. Ahner, LTC, USA (Chairman)	Date
________// signed //____________________	__6 Mar 12__
Jeffrey D. Weir, PhD, (Member)	Date

AFIT/OR-MS/ENS/12-09

Abstract

With the advancing capabilities of Intelligence, Surveillance, and Reconnaissance (ISR) assets and sensors, effective utilization of these resources continues to pose a challenge to military decision makers. The methodology developed explores allocation of ISR assets while balancing detection of new targets versus surveillance of already detected targets (discovery vs. persistence) using entropy as a measure of effectiveness. Scenarios with an unknown number of static and moving targets in a bounded geographical region are considered. A baseline model was built to examine four different search algorithms: random, raster, greedy, and a rollout algorithm based on dynamic programming. A space-filling Nearly Orthogonal Latin Hypercube (NOLH) experimental design was applied to generate data to examine four Measures of Effectiveness (MOEs): step entropy, average entropy, number of targets found, and time steps to completion.

Based on statistical analysis and time series plots, the rollout algorithm's performance dominated others algorithms considered for all MOEs. In addition to minimizing uncertainty in the first 100 time steps of the run, the rollout algorithm also produced the highest number of targets found within the fixed time step scenario, and, for the exhaustive target detection scenario, discovered all of the targets within the region in less time steps. Based on these results, the rollout algorithm provides superior performance in the allocation of ISR assets while balancing detection of new targets versus surveillance of already detected targets.

AFIT/OR-MS/ENS/12-09

To my husband, and our unconditionally loving “furry menagerie”

Acknowledgments

I would first like to thank my husband for being the best support I could have asked for during the past 18 months. Also, for all of the words of encouragement from my family and friends. I could not have done this without all of you.

I would like to express my sincere appreciation to my faculty advisor, LTC Darryl K. Ahner, for his guidance and support throughout the course of this thesis effort. The continual pushes in the right direction, as well as the ever-flowing Army vs. Air Force jokes were always appreciated.

A special "thank you" also needs to go out to the entire ENS Section 2 …my success in this program was a direct correlation to how awesome you all were. And more importantly, The Price is Right would not have had such a profound effect on my AFIT experience and life.

Tamilyn S. Dismukes

Table of Contents

Page

List of Figures

List of Tables

Page

SURVEILLANCE VERSUS RECONNAISSANCE:

AN ENTROPY BASED MODEL

I. Introduction

The former United States Secretary of Defense, Dr. William J. Perry, stated, “We live an age that is driven by information. Technological breakthroughs…are changing the face of war and how we prepare for war”. With the increase of information and the ever-tightening military budget, the Department of Defense (DoD) is continually looking to defense planners, commanders, and decision-makers to make intelligent decisions regarding the use of Intelligence, Surveillance, & Reconnaissance (ISR) assets during war and peacetime. This thesis examines the use of information superiority and the growing need to effectively utilize ISR assets within the battle space to meet our national security needs.

Background

With the advancing capabilities of ISR assets and sensors, effective utilization of these resources continues to pose a challenge to military decision makers. There are numerous questions that a decision maker can ask in regards to this challenge. What quantity and mix of ISR assets are needed to meet our national security challenges? How should ISR assets be used? What quantity and mix of surveillance and reconnaissance force is needed to cover a defined area? How do we measure our current ISR capabilities? As these questions are explored, two overarching measures of merit have been defined: discovery of new targets (reconnaissance) and persistence of already known targets (surveillance). Both are important to the situational awareness of the battle

space, however, “we can optimize discovery or we can optimize persistence…but we cannot do both” (Murphy & Payne, 2009). Tradeoffs between discovery and persistence should be explored and methodologies created that aid this exploration.

Problem Statement

This research examines search algorithms for an unknown number of static and moving targets over a discrete time and bounded domain for ISR assets. The methodology explored will balance detection of new targets versus surveillance of already detected targets (discovery vs. persistence) using entropy as a measure of effectiveness.

Preview

Chapter II explains background information and previous research on entropy, information gain, and common search algorithms. Chapter III details the implementation of certain heuristics in an entropy based model of a defined ISR battle space. Chapter IV includes the analysis and results of this method. Finally, Chapter V provides an overview of the work completed in this paper, as well as recommendations for future work.

II. Literature Review

This chapter outlines background information and previous research on entropy, information gain, and common search algorithms. This chapter is organized into 6 sections: Bayesian Updating, Entropy, Information Gain, Diffusion Modeling, Entropy Measure of Surveillance Effectiveness, and Search Algorithms.

Bayesian Updating

Bayesian updating is commonly used in the formulation surveillance operations to determine the existence or non-existence of targets within a region. The cells are initialized using an a priori probability, or initial degree of belief that a target is in a specific cell of the region, P(A). An a posteriori probability, the degree of belief when information of a target detection or non-detection occurs P(A|B), is then determined through the application of Bayes' Theorem in Equation 1 below. Assuming all cells of the region are independent allows for the application of Bayesian updates to only those cells that are being observed by the surveillance sensor. Bayesian updates are not applied to those cells outside of the sensor detection since no new information is received. (Berry, Pontecorvo, & Fogg, Optimal Search, Location and Tracking of Surface Maritime Targets by a Constellation of Surveillance Satellites, July, 2003)

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{1}$$

Entropy

Shannon's concept of entropy is the basis for modern information theory. It was originally established as a measure of the information contained in a transmitted message.

It is often referred to as the "measure of uncertainty", and is a numerically measurable quantity, on the basis of a probabilistic model. In Shannon's entropy derivation, the outcome space, Ω, includes a discrete number of mutually exclusive outcomes or events, X_i. For each outcome, X_i, there corresponds a probability of occurrence, p_i. The function $H_n(p_1, p_2, \ldots p_n)$ is to be interpreted as the average uncertainty associated with the outcome $X=x_i$, $i = 1, 2, \ldots, n$. Equation 2 below is Shannon's original entropy equation.

$$H_n(p_1, p_2, \ldots p_n) = -K \sum_{i=1}^{n} p_i \log(p_i) \qquad (2)$$

Where K = positive scaling constant

p_i = probability of outcome i

Entropy is traditionally unitless. The higher the entropy value, the greater the uncertainty of the information received. An entropy value of zero corresponds to no uncertainty of information, i.e. perfect information. The higher the entropy value, the greater the uncertainty in the information at hand. (Shannon, 1949)

Shannon's entropy displays a number of desirable properties, four are outlined below. (Reza, 1961)

(1) *Continuity*. The entropy function is continuous in each and every independent variable p_i.

(2) *Symmetry*. The entropy function is symmetric for every combination of probabilities p_i.

(3) *Extremal Value of the Entropy Function*. The entropy function has a maximum value when all the individual probabilities are equal. This maximum value varies based on the number of mutually exclusive outcomes.

(4) *Additivity*. The total entropy of a sample space is equal to the sum of its parts

Information Gain

Barr and Sherrill (Barr & Sherrill, July, 1996) applied entropy when considering a concept of information gain. Information gain measures the decision maker's state of uncertainty about his adversary in terms of discrete probability distributions over space that the adversary may occupy. Their question was, "How does a commander's state of knowledge change when he receives new data containing information?" Their research modeled the amount of uncertainty a commander faces when applied to a scenario with a finite set of possible states and a probability distribution over the set which may be updated as information is received.

Like entropy, information gain is unitless because the measure depends only on the probabilities of the possible outcomes. Once the entropy at a given state is calculated, a Bayesian update is used to determine the new probability, given that a sensor detects the target. This probability is used to calculate the new entropy. The difference between the old entropy and the new entropy is the "information gain", shown in Equation 3 below.

$$\delta(p, p^*) = -\sum_{i \in S} [p_i \ln(p_i) + p_i^* \ln(p_i^*)] \tag{3}$$

Where p = prior distribution of the commander's uncertainty

p^* = posterior distribution of the commander's uncertainty at a later time

This information gain in a military context addresses a primary objective of "studying the relationship between information gained about an enemy's disposition and measures of combat effectiveness (Barr & Sherrill, July, 1996)". However, this information gain term needs to be normalized across competing requirements (Ahner, 2009 Winter Simulation Conference).

Diffusion Modeling

Probability diffusion is defined as the spreading out of a probability throughout a state space over time. A probability diffusion model represents the rate of transition of that spread. This model is commonly used to represent the probability of a moving target location over time without additional updates on the target's location. A diffusion model must follow the following axioms of probability as stated in Montgomery and Runger's "Applied Statistics and Probability for Engineers" (Montgomery & Runger, 1999):

(1) All outcomes must have a probability between zero and one inclusive:

$0 \geq P(X = x_i) \leq 1 \ \ for\ all\ i$

(2) The sum of all the outcomes must equal one:

$\sum_i P(X = x_i) = 1$

(3) If two outcomes cannot happen at one time, the probability is derived from the addition of the individual probabilities:

$P(X_1 \cup X_2) = P(X_1) + \ P(X_2), \qquad given\ X_1 \cap X_2 = \emptyset$

Shupenus and Barr (Shupenus & Barr, 1999) discuss three types of target diffusion models: the square uniform, the circular uniform, and the exponential cone model.

Square Uniform Model

The square uniform model assumes that at a specified time increment a target is equally likely to be found in its last known location, as well as each of the adjacent cells. Figure 1 below is a visual representation of the square uniform diffusion model. This model allows for easy probability computation. There are, however, a number of assumptions that create an unrealistic target movement characterization. The model assumes a target could travel to the extreme corner of a cell in the same time that it can reach the middle. In addition, the model does not regard target speed.

0	0	0	0	0
0	1/9	1/9	1/9	0
0	1/9	1/9	1/9	0
0	1/9	1/9	1/9	0
0	0	0	0	0

Figure 1: Square Uniform Diffusion Model
(Shupenus & Barr, 1999)

Circular Uniform Model

With the circular uniform model, the number of cells a target can reach in a given time is limited based on speed and the potential distance traveled. Figure 2 below is a visual representation of the circular uniform diffusion model. This model is also

unrealistic, in that a target is equally likely to be at the edge of the circle as it is to be at its last know location. This is not the case, even in a random walk.

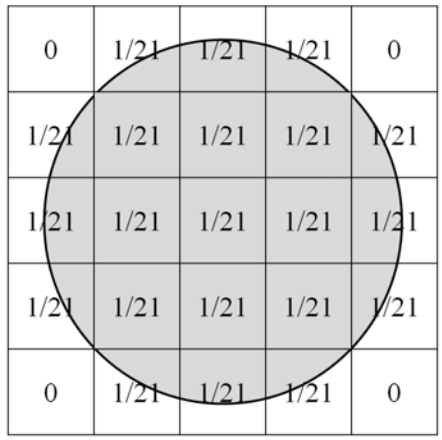


Figure 2: Circular Uniform Diffusion Model
(Shupenus & Barr, 1999)

Exponential Cone Model

The exponential cone model characterizes the target's movement with the bivariate distribution. Figure 3 below is a visual representation of this distribution, which has the shape of a curved cone, hence the name exponential cone diffusion model. This model is a better representation of typical target movement than the uniform models above because it is expected that there is a greater likelihood of finding the target somewhere near the last know location, than at the farthest possible point based on the max speed of the target. The likelihood decreases from a maximum value at the center, to zero at the maximum radius. The rate of diffusion should be determined based on target characteristics, such as speed and initial location, as well as battlefield characteristics, such as terrain and topography.

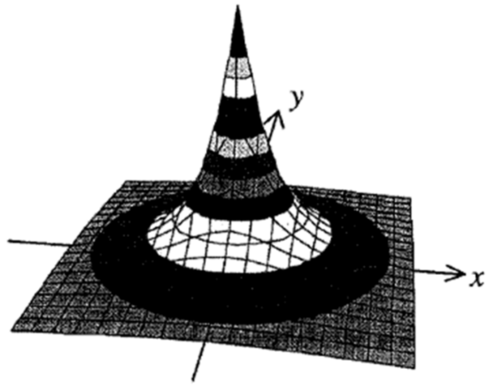


Figure 3: Exponential Cone Diffusion Model
(Shupenus & Barr, 1999)

Entropy Measure of Surveillance Effectiveness

A recent study by Berry, Pontecorvo and Fogg (Berry, Pontecorvo, & Fogg, July, 2003) facilitates the optimal solutions to dynamically determining the allocation and control of satellite surveillance resources for the purpose of detecting, locating, and tracking surface maritime targets. The formulation was based upon Bayesian estimation and an entropy measure of surveillance effectiveness. A sensor's performance was modeled in terms of its probability of detection (p_d) and false alarm (p_{fa}). Surveillance operations were broken down into three separate phases: search, locate, and track. For each phase, the objective was to maximize the expected information. These phases are described below.

Phase I: Search

During the search phase, the objective was to determine the existence or non-existence of targets within a region. Targets and the cells of the region were assumed to be independent of each other, and the probability that a target was located in a cell was subjected to Bayesian updates based on whether or not detection occurred. The a priori

probability for a target cell i at epoch $n+1$ was p_i^{n+1}. Equation 4 below denotes the a posteriori probability if a detection did occur and Equation 5 below denotes the a posteriori probability if a detection did not occur.

$$p_i^{n+1}(1) = \frac{p_d \hat{p}_i^{n+1}}{p_d \hat{p}_i^{n+1} + p_{fa}(1 - \hat{p}_i^{n+1})} \tag{4}$$

$$p_i^{n+1}(0) = \frac{(1 - p_{d)} \hat{p}_i^{n+1}}{(1 - p_{d)} \hat{p}_i^{n+1} + (1 - p_{fa)}(1 - \hat{p}_i^{n+1})} \tag{5}$$

The entropy for the global distribution of targets was equal to the sum of all of cells in the region. The entropy of a cell i at epoch $n+1$, with a non detection, was denoted as $(h_i^{n+1}(0))$ The entropy of a cell i at epoch $n+1$, with a detection, was denoted as $(h_i^{n+1}(1))$. The individual entropy for each separate cell, based on whether a detect or non detect occurred, was calculated using Equation 6 and Equation 7 below.

$$h_i^{n+1}(0) = -p_i^{n+1}(0) log p_i^{n+1}(0) - \left(1 - p_i^{n+1}(0)\right) \log\left(1 - p_i^{n+1}(0)\right) \tag{6}$$

$$h_i^{n+1}(1) = -p_i^{n+1}(1) log p_i^{n+1}(1) - \left(1 - p_i^{n+1}(1)\right) \log\left(1 - p_i^{n+1}(1)\right) \tag{7}$$

Hence, the expected entropy is

$$\mathrm{E}(h_i^{n+1}) = \Pr\{0\} h_i^{n+1}(0) + \Pr\{1\} h_i^{n+1}(1) \tag{8}$$

Where $\Pr\{1\}$ = probability of detection

$\Pr\{0\}$= probability of no detection

The entropy for the global distribution of targets was equal to the sum of the entropies for the individual cells. Therefore, a sensor action was selected to minimize the total expected entropy of the region (Berry, Pontecorvo, & Fogg, July, 2003).

Phase II: Locate

During the locate phase, the objective was to determine the location of the targets discovered during the search phase. A probability distribution for each of the predicted number of k targets is determined with using Bayesian updates of observations in cells $\{j_1, j_2, \dots j_k\}$. Therefore, the probability distribution for the locations of k targets at the nth epoch is $P^n(j_1, j_2, \dots j_k)$. The entropy corresponding to the information regarding the locations of the predicted number k of targets at the n^{th} epoch is

$$h^n = -\sum_{j_1 \cdots j_k} P^n(j_1, j_2, \dots j_k) \log P^n(j_1, j_2, \dots j_k) \tag{9}$$

The choice of sensor control parameter was based upon the expectation of entropy change following the observations of a target. (Berry, Pontecorvo, & Fogg, July, 2003)

Phase III: Track

During the track phase, Bayesian updates and a Markovian target motion model were used to track target location estimates. Each target maintained a separate probability distribution for its location and the sensor was tasked to track a single target of interest (Berry, Pontecorvo, & Fogg, July, 2003).

The above search and track methods were based on the posterior measure of probability. However, the method of search to find the a priori probability was not discussed. The next section discussed various types of search algorithms that are used to determine those values.

Search Algorithms

Exhaustive Algorithms

There are several exhaustive search algorithms. The “raster scan” method, the “spiral in” method, and the “spiral out” method, are the most common. The raster scan method sweeps vertically (or horizontally) over the region of interest, and is depicted in Figure 4a below. The spiral in method starts the sweep on the outer edge of the region and spirals in towards the center, and is depicted in Figure 4b below. The spiral out method starts the sweep in the center of the region and spirals out towards the outer edge, and is depicted in Figure 4c below. Given that the target is stationary and equally likely to be in a given cell, all of these methods are equivalent. However, if targets move as the search is carried out, or the probability of detect is less than 1, these methods become less favorable (Washburn, 2002).

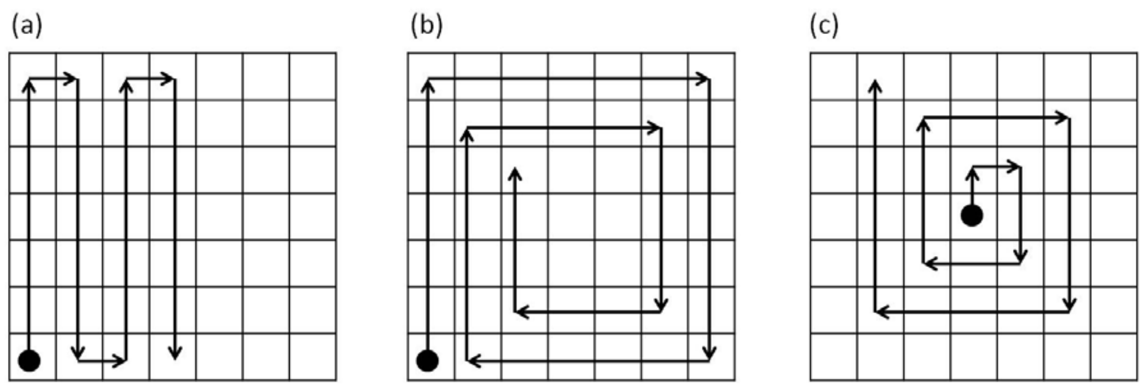


Figure 4: Exhaustive Search Algorithms
(a) Raster Scan (b) Spiral In (c) Spiral Out

Greedy Algorithm

A greedy algorithm always makes the choice that looks best at that moment. It progresses in a top down manner, making one greedy choice after another. It chooses a local optimal in hope that this choice will lead to a globally optimal solution. This

heuristic strategy rarely produces an optimal solution, but can on occasion. There is no general way to determine whether a greedy heuristic will produce an optimal answer, but there are two properties that support the use of such an algorithm: the greedy-choice property and the optimal-substructure property (Cormen, Leiserson, & Rivest, 1989).

The greedy-choice property is defined as a problem in which a globally optimal solution can be arrived at by making a locally optimal choice. In order to determine this property, it must be proven that a greedy choice yields a globally optimal solution at each step, which can be difficult. A problem possesses the optimal-substructure property if an optimal solution to the problem contains within it optimal solutions to subproblems (Cormen, Leiserson, & Rivest, 1989).

Rollout Algorithm

Rollout algorithms are commonly used in combinatorial problems such as scheduling and routing. They were first proposed for the approximate solution of discrete optimization problems by Bertsekas, Tsitsiklis and Wu (Bertsekas, Tsitsiklis, & Wu, Rollout Algorithms for Combinatorial Optimization, 1997), and are capable of magnifying the effectiveness of any given heuristic algorithm through sequential application.

The problem set is characterized by a finite set U of feasible solutions and by a cost function $g(u)$. The problem is viewed as a sequential decision problem where the components $u_1 \ldots u_N$ are selected one-at-a-time. The initial state is an empty set of decisions, where an n-solution is formed consisting of the first n decisions. From this state, a decision u_{n+1}, is added to form the (n+1) solution where

$$U_{n+1} = \{u_{n+1} | \text{ there exists a solution of the form } (u_1, u_2, \ldots, u_N) \epsilon U\} \quad (10)$$

$J^*(u_1, u_2, \ldots, u_N)$ denotes the optimal cost starting from the n-solution. If J^* is known, the optimal solutions can be constructed through a sequence of N minimizations. Unfortunately, J^* is rarely known and every possible permutation of the decision space would have to be explored to find it. To deal with this, an approximation $\tilde{J}(\tilde{u}_1, \tilde{u}_2, \ldots \tilde{u}_N)$ to obtain a suboptimal solution, $(\tilde{u}_1, \tilde{u}_2, \ldots \tilde{u}_N)$, is applied through Equation 11 below.

$$u_i^* = \arg\min_{\tilde{u}_i \epsilon U_i} J(\tilde{u}_1^*, \ldots \tilde{u}_{i-1}^*, \tilde{u}_i^*) \quad i = 1 - N \quad (11)$$

Heuristic algorithms are used to obtain the approximate cost to go function, $\tilde{J}(\tilde{u}_1, \tilde{u}_2, \ldots \tilde{u}_N)$, by starting with an n-solution whose cost is denoted by $H(u_1, u_2, \ldots, u_N)$. Then, $\tilde{J}(\tilde{u}_1, \tilde{u}_2, \ldots \tilde{u}_N) = H(u_1, u_2, \ldots, u_N)$, as the approximate cost to go (Bertsekas, Tsitsiklis, & Wu, Rollout Algorithms for Combinatorial Optimization, 1997)

Rollout algorithms have been shown to significantly improve performance of index and greedy heuristics, and are computationally tractable. The rollout algorithm is also monotonically increasing (Bertsekas & Castanon, Rollout Algorithms for Stochastic Scheduling Problems, 1999).

III. Methodology

This chapter outlines the methodology and experimental design setup for the model used to for this research. This chapter is organized into 11 main sections discussing grid characteristics, target characteristics, asset characteristics, entropy growth, diffusion modeling, Design of Experiments (DOE), and Measures of Effectiveness (MOEs).

Hexagonal Coordinate System

A hexagonal coordinate system was used for this simulation. One of the major advantages to this type of system lies in the consistent connectivity of its constituent hexagons. All adjacent hexagons are equidistant to the center hexagon, depicted in Figure 5a below. This allowed for ease of computation in both time steps and the diffusion model used and improves upon the diffusion models described in Chapter II.

A conversion of the hexagonal coordinate system to the traditional Cartesian coordinate system was needed for coding and animation in Visual Basic. Figure 5b below is a representation of this mapping (Hexagonal Coordinate Systems, 2005).

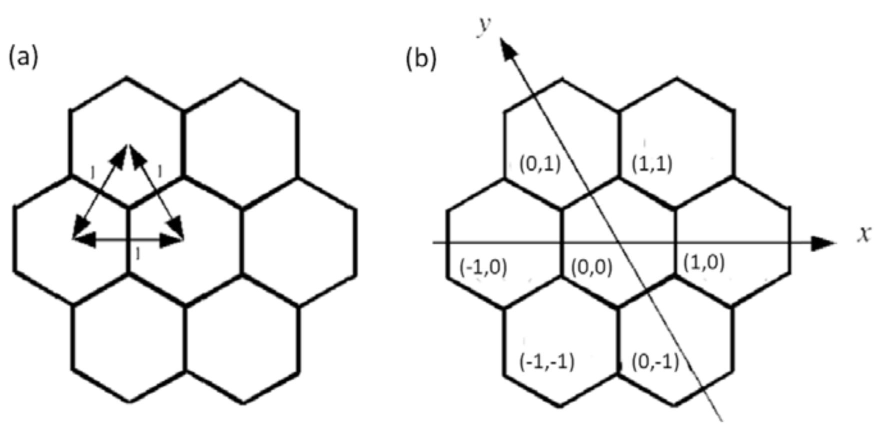


Figure 5: Hexagonal Coordinate System
(a) Equidistant Hexagons (b) Cartesian Coordinate Mapping

Grid Characteristics

A rectangular grid was constructed based on the size of the area of interest. Figure 6 below is a visual representation of that grid. Each hex was assumed to be independent within the grid. This allowed for computational ease for entropy and target diffusion.

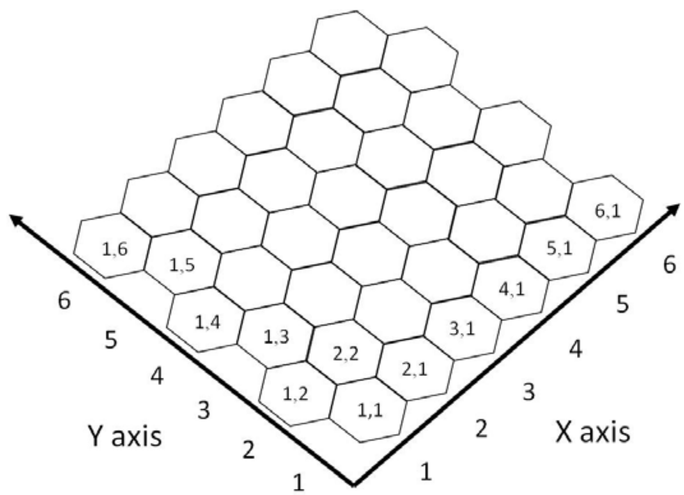


Figure 6: Rectangular Grid

Target Characteristics

Targets were separated into two categories: stationary and moving. Stationary targets remained in one hex for the entirety of the simulation. Moving targets had the ability to travel in seven possible ways each time step: move to one of the six adjacent hexes, or stay stationary. The targets mix was changed based on each scenario.

A number of assumptions were used in regards to target characteristics:

(1) Moving targets chose their directional movement randomly from a maximum of seven different movements: to stay stationary, or to move to one of six adjacent hexes, within the grid boundaries. No ground truth involving roads, terrain, or structures was utilized.

(2) Targets moved at a 1:1 ratio in reference to the speed of the ISR asset.

(3) Each hex contained a maximum of one target during each time step.

(4) Each target had a unique identifier. An asset could distinguish between a new target and a target already identified.

Asset Characteristics and Movements

An asset was defined as an aerial based ISR platform containing a sensor with the capability to search and track stationary and moving targets on the ground. A homogeneous asset force was used, therefore, the hex sizes in the grid were based on sensor platform capability. This sensor platform capability was one of the user inputs. Time steps were based on the ability of the asset to move one time per time step.

Each asset started from the same location on the grid and had the ability to travel in seven possible ways within the grid: move to one of the six adjacent hexes, or stay in the same location. These movements were determined by the movement algorithm selected for each scenario. These movement algorithms are outlined below.

Random Movement

The random movement algorithm was the simplest movement algorithm used. The travel direction of the asset was chosen randomly with the grid boundaries as the only constraint. This algorithm was used as a lower bound on algorithm performance.

Raster Movement

The exhaustive search method used in this simulation was the raster movement algorithm. This algorithm moved the asset in a pre-determined up and down sweeping motion within the grid boundaries. Figure 7 below is a visual representation of this

movement. This algorithm performs favorable when targets are stationary and the probability of detect is one. It was used as an upper bound on algorithm performance when these conditions were met.

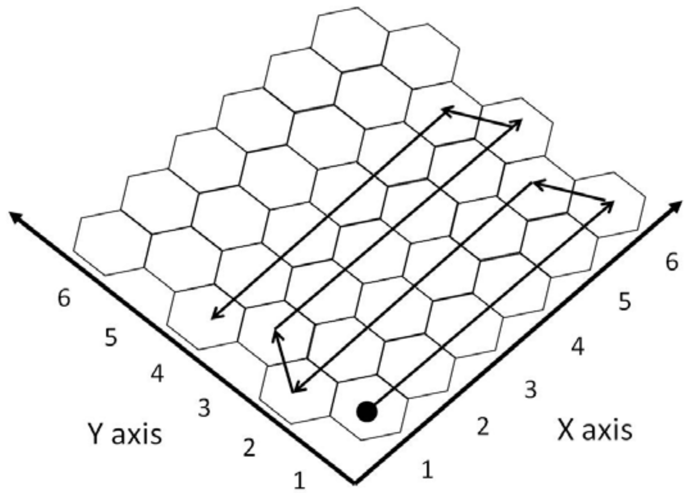


Figure 7: Raster Algorithm Example

Greedy Heuristic Movement

The greedy heuristic algorithm determined the travel direction of the asset by choosing the direction of travel to the adjacent hex that had the highest entropy value within the grid boundaries. Figure 8 below is a visual representation of a greedy movement example. Based on the entropy values in the figure, the asset would choose to move to the hex with the highest entropy value of 0.5. If more than one hex contained the highest entropy value of the adjacent hexes, the travel direction was chosen randomly among those hexes.

Movement Path	Entropy Value	Path Chosen
⟶	0.2	Yes
⋯⋯→	0.1	No
- - →	0.1	No

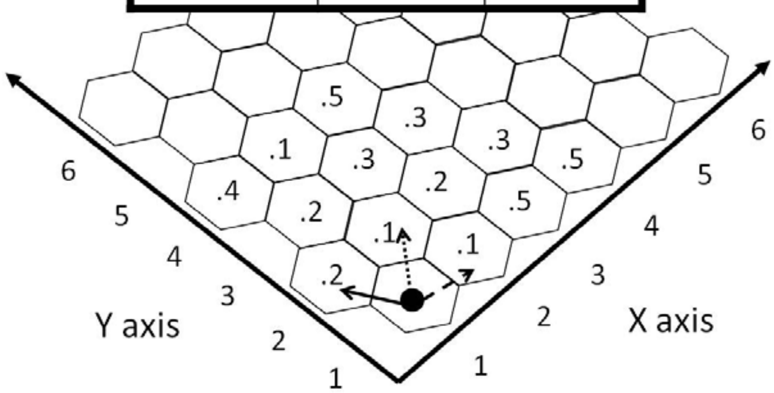


Figure 8: Greedy Algorithm Example

Rollout Heuristic Movement

The rollout heuristic algorithm determined the travel direction of the asset by utilizing the rollout algorithm discussed in Chapter II. This algorithm chose the path with the highest overall entropy according to the corresponding number of movement look aheads. Figure 9 below is a visual representation of a rollout movement example with two look aheads. Based on the entropy values in the figure, the asset would to move to the hex with the highest path entropy value of 1.1. If more than one path contained the highest entropy value, the travel direction was chosen randomly among those paths.

Movement Path	Path Entropy	Path Chosen
- - - ->	0.2+0.2+0.4 = 0.8	No
············>	0.1+0.3+0.5 = 0.9	No
———>	0.1+0.5+0.5 = 1.1	Yes

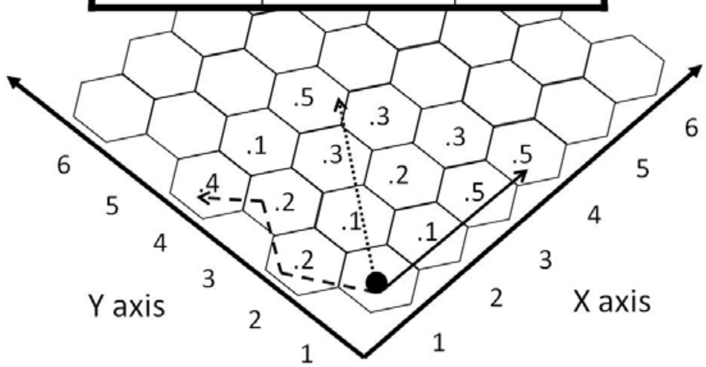


Figure 9: Rollout Algorithm Example

Track Movement

During some scenarios, an asset would be used to track a target instead of search the grid. An asset designated for this purpose would follow the specified search algorithm until a target was identified for track. At this point, the asset would switch to the track movement and follow that specified target. As stated in the target characteristics section above, at 1:1 ratio was assumed between the asset and target speeds.

Multiple Asset Movements

Additional rules were applied to the asset movements when more than one asset traveled within the grid. Below, these rules are explained according to the movement algorithm selected.

Raster Movement Scenario

During the raster movement scenario, only one asset started movement at the first time step. Each additional asset was delayed a specified amount of time to evenly

distribute the assets over the grid. The time delay was calculated as the total amount of grid hexes divided by the total amount of assets. As the assets swept back and forth over the grid, there were instances when the assets occupied the same hex. It was assumed that there was enough altitude separation for this to occur.

Random, Greedy, and Rollout Movement Scenarios

During the random, greedy and rollout movement scenarios, the only time assets occupied the same hex was at the start of the scenario. Once an asset departed the base, it was not allowed to occupy the same hex as another asset. This was done to allow for maximum coverage of the grid.

Search versus Track

Assets also had the ability to be designated a track asset as one of the model inputs. Asset followed the search algorithm selected for the run until a moving target was detected by that asset. If the number of assets tracking targets was less than the number designated to track, the asset stopped searching and tracked that specific target. The probability of keeping that track was also a model input. If the asset lost the track of the target, the asset resumed the search algorithm until another moving target was identified.

Entropy Growth

The entropy values for each cell of the grid varied from zero, representing no uncertainty, to 0.5, representing the maximum value of uncertainty. This normalization allowed for comparative analysis between the runs. Over time, the entropy for each hex degraded from the initial entropy value to the maximum value of 0.5. This growth was derived using the generalized harmonic stepsize function:

$$Stepsize = a/(a+n) \tag{12}$$

Where a = positive constant

n = time step

Increasing the value of *a* slows the rate at which the stepsize drops to zero, as the entropy growth reaches a value of 0.5 (Powell, 2007). Figure 10 below is a visual representation of the entropy growth for varying values of *a*.

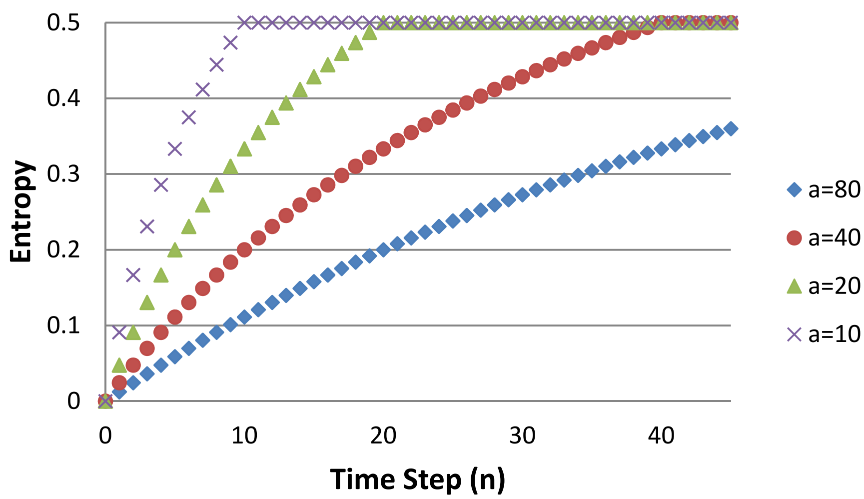


Figure 10: Generalized Harmonic Entropy Growth

This growth function was chosen because of its flexible growth rates. The value of *a* was altered depending on what type of target was identified in that cell. The growth rate of a hex with a stationary target would be less than that of a moving target. A growth rate of 80 and 10 were used, respectively, for hexes with stationary and moving targets.

Belief State Diffusion

As the ISR asset traveled within the grid, a belief state on target location probability was built. This belief state was represented as a two dimensional matrix with

the dimensions of the grid. Each cell of this belief state matrix contained the probability of a target being located in the corresponding grid hex. When a target was detected in a hex, the corresponding belief state cell was updated to a probability of 1 (since the probability of a false alarm was assumed to be zero). If the target was a moving target, the belief state incorporated a diffusion rate to represent the probability of a moving target over time without any additional updates on the target's location.

The diffusion model used for the belief state incorporated the uniform square model and the exponential cone model discussed in Chapter II. The uniform square model was chosen based on its computational ease and the exponential cone model was chosen based on its more realistic representation of target movement. Moving targets chose their directional movement randomly from a maximum of seven different movements: to stay stationary, or to move to one of six adjacent hexes, within the grid boundaries. Therefore, each of the seven moves was modeled with equal probability.

These moves were modeled over discrete time steps, creating a belief state diffusion over time. Figure 11 below is a visual representation of this diffusion for one target over 3 time steps, using an 8 direction square grid. Figure 11a depicts an initial detect of a moving target at cell (4,4) with probability 1. Figure 11b depicts the first time step after initial detection. A moving target is equally likely to stay in its initial cell or move to one of the adjacent cells, therefore the probability of a target in cell (4,4) and each adjacent cell is 1/9. Figure 11c depicts the second time step after initial detection. Those cells containing probabilities in Figure 11b are diffused in the same pattern. Since cell (4,4) is adjacent to all of these diffusing cells, the probability that a target is in cell (4,4) is more likely than those cells on the edge of the diffusion pattern. As the time steps

increase, the diffusion model starts to represent a bivariate normal distribution, where the probability of a target in a cell is more likely in the center, than on the edges.

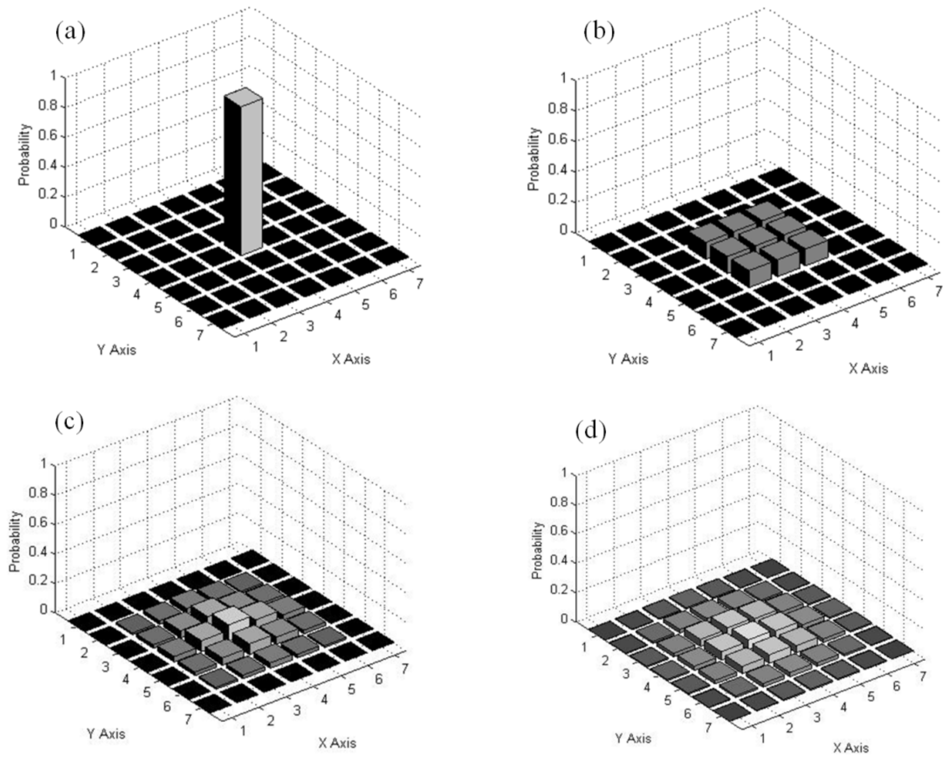


Figure 11: Diffusion Model Time Steps
(a)Time Step Zero (b)Time Step One (c)Time Step Two (d)Time Step Three

The belief state diffusion continued over time until one of two events occurred. The first event was an additional detection of an already known target. This was characterized as an update to that target location and the belief state was refreshed to time step zero in order to start the diffusion process over. The second event that halted the belief state diffusion was if the maximum diffusion probability for a target reached a specified tolerance. If the maximum diffusion probability fell below the tolerance level,

a uniform distribution is applied to the entire grid for that target, capturing that the target was identified; however, there was no knowledge on where that target currently is. This distribution does not vary until the target was detected again.

Since each target had a unique identifier, each target had an individually calculated belief state. Based on the axioms of probability, the expected number of targets is the sum over all the belief states.

$$E(Number\ of\ Targets) = \sum\nolimits_{i} Belief\ State_i \qquad (13)$$

Where i = Number of identified targets

Model Inputs

The model called for several user inputs. These inputs includes size of the grid, number of stationary targets, number of moving targets, number of search assets, number of track assets, the starting location of those assets, the size of the sensor view, the sensor probability of detect, the probability an asset could keep a track, as well as the number of "look aheads" if the rollout algorithm was in use.

Design of Experiments

When conducting simulation studies, there are typically a large number if inputs and a finite amount of computing resources available to perform simulation runs. Design of Experiments provides a structured way to decide which configurations to simulate so that the desired output is obtained in the fewest possible runs. The primary goal of experimental design is to assess how changes to input parameters (factors) affect the results (responses) of the simulation.

Factor Selection

Based on the model inputs outlines in the section above, the factors selected for the DOE fell into three categories: asset/sensor characteristics, target characteristics, and search algorithm parameter. The first set of factors included the following asset/sensor characteristic: number of track assets, number of search assets, sensor probability of detect (P_{Det}), and sensor probability of track (P_{Track}). The second set of factors included the following target characteristics: number of stationary targets and number of moving targets. The final factor was the number of look aheads used when the rollout algorithm was set as the search type. These factors were selected base on their possibility of being an influential factor. Table 1 below outlines the factors and their ranges used in the DOE.

Table 1: Design Factors and Ranges

Design Factor	Factor Ranges	
	Low	High
Number of Track Assets	0	5
Number of Search Assets	1	5
Number of Stationary Targets	1	30
Number of Moving Targets	1	30
Sensor Probability of Detect	0.20	1.0
Sensor Probability of Track	0.20	1.0
Number of Look Aheads	2	8

Nearly Orthogonal Latin Hypercubes

After input factor selection, an experimental design was selected to provide the responses required to fully explore the range of input factors. The space-filling Nearly Orthogonal Latin Hypercube (NOLH) design was selected to provide an exploration of the entire response surface. The NOLH design allows for multiple levels, or even continuous ranges, for each factor (Cioppa & Lucas, July, 2007). Another advantage of

the NOLH design is the minimization of the correlation between the columns of the design matrix to produce a “nearly orthogonal” design, which is key property of a good design (Kleijnen, Sanchez, Lucas, & Cioppa, 2005).

A spreadsheet design tool, was used to generate the design points for the seven factor NOLH design (Sanchez, 2005). The spreadsheet employs an algorithm where the maximum number of factors examined in a Latin hypercube is $m + \binom{m-1}{2}$ where m is an integer greater than 1. Solving for m using 7 as the number of factors gives $m = 4$. The number of n design points required is given by $n = 2^m+1$ and results in 17 design points needed (Cioppa & Lucas, July, 2007). These design points were run for each of the search algorithms explained above. Table 2 below outlines the factor levels for each of the 17 design points in the DOE.

Table 2: Nearly Orthogonal Latin Hypercube Design Points

Factor	# Search Assets	# Track Assets	Probability Detect	Probability Track	# Stationary Targets	# Moving Targets	# Look Aheads
Low Level	1	0	0.20	0.20	1	1	2
High Level	5	5	1.0	1.0	30	30	8
Design Points	Design Point Levels						
1	5	2	0.40	0.95	25	12	5
2	2	0	0.20	0.45	26	17	6
3	3	1	0.70	0.85	3	8	8
4	4	1	0.65	0.30	10	30	7
5	5	4	0.45	0.20	14	5	7
6	2	5	0.25	0.80	12	25	7
7	2	3	0.90	0.55	30	10	8
8	5	3	0.85	0.70	23	28	6
9	3	3	0.60	0.60	16	16	5
10	1	3	0.80	0.25	6	19	5
11	4	5	1.00	0.75	5	14	4
12	3	4	0.50	0.35	28	23	2
13	3	4	0.55	0.90	21	1	4
14	1	1	0.75	1.00	17	26	3
15	4	0	0.95	0.40	19	6	3
16	4	2	0.30	0.65	1	21	2
17	2	2	0.35	0.50	8	3	4

Scenarios

The NOLH design described above was applied to two different scenarios. These scenarios differed in the stopping criteria for the simulation. The first scenario was run until all of the targets within the grid were detected, and was called the exhaustive target detection scenario. The second scenario was run until a fixed number of time steps was reached, and was called the fixed time step scenario. The Measures of Effectiveness (MOEs) for each scenario are outlined below.

Common Scenario MOEs

The two common MOEs used during the exhaustive target detection and fixed time step scenarios were the step entropy and average entropy.

Step Entropy

The step entropy of the region was generated by the model after each time step of the run. The step entropy was calculated using Equation 14 below. This MOE was selected as it was an appropriate measure to provide insight into the trends of entropy over time for each run.

$$Step\ Entropy_i = \sum_j \sum_k Entropy_{jk} \tag{14}$$

Where i = Time Step

j = Number of hexes in the X axis

k = Number of hexes in the Y axis

Average Entropy

The average entropy of the region was generated by the model for each run. The step entropy was calculated using Equation 15 below. This MOE was selected as it was

an appropriate measure to provide insight into the overall uncertainty of information over the entire run.

$$Average\ Entropy = \frac{(\sum_i\ StepEntropy_i)}{TotalTime} \tag{15}$$

Where i = Time Step

Exhaustive Target Detection Specific MOE

Each design point within the exhaustive target detection scenario ran until all of the targets were detected, therefore, the number of time steps varied based on the search algorithm utilized and target movement. This MOE was selected as it was an appropriate measure to provide insight into the efficiency of the search algorithms in the model.

Fixed Time Step Specific MOE

Each design point within the fixed time step scenario ran until a specified number of time steps were completed, limiting the target detection within the region, therefore, the number of targets detected varied. The number of time steps was set at 700 for this scenario. This number was chosen to allow the model to reach a steady state entropy value for each of the search algorithms. This MOE was selected as it was an appropriate measure to provide insight into the efficiency of the search algorithms in the model.

IV. Results and Analysis

This chapter outlines the results and analysis of the output generated from this research. The chapter is organized into six main sections: Five sections discussing each of the MOEs (step entropy, average entropy, number of targets found, number of time steps to completion, and surveillance vs. reconnaissance), and the fifth discussing overall conclusions.

For all of the results, the region of interest was held constant with an 80 km by 80 km grid. These values lead to upper and lower entropy bounds on the region. The upper entropy bound was determined with all of the hex values at the maximum entropy value of 0.5. The lower entropy bound was determined by utilized a design run that would results in the lowest possible entropy level within the model. This run consisted of a P_{Det} of 1.0, ten assets, and all stationary targets. The upper bound was 418, and the lower bound was approximately 250.

Step Entropy Results

The step entropy MOE was generated by the model for each design point of the NOLH using Equation 14. The entropy of the region was plotted over time to provide insights into the trend of entropy within the run. All of these plots contained one common trend: the rollout search algorithm decreased the entropy of the region the most during the first 100 time steps, and continued to remain as one of the lowest entropy values throughout the run. The greedy algorithm behaved in a similar manner, while the raster and random algorithms did not have a clear trend among all of the runs. The

resulting plots for each of the 17 design points are located in Appendix A. A subset will be discussed in detail below.

The plots can also be separated into two groups of behavior: those that reached a steady state with an entropy value below 400, and those that had an entropy value that did not drop below 400. The main factor that separated these two groups was the probability of detect. Those design runs with a high P_{Det} (≥ 0.85) all belonged to the first group. The remaining design runs ($0.2 \leq P_{Det} < 0.85$) belonged to the second group. This separation of design points is expected since a sensor with a higher P_{Det} leads to a less uncertain environment. The trends within these two groups are discussed below.

Steady State Entropy Below 400

The four design points that produced runs which fell into the first group of behavior (runs that reached a steady state with an entropy value below 400) were design points 7, 8, 11, and 15 (outlined in Table 2). Figure 12 and Figure 13 below are the plots for the runs of design points 7 and 11, respectively. The entropy at time step 0 always started at a value of 418.5 due to the initial state of the grid containing an entropy value of 0.5 in each hex.

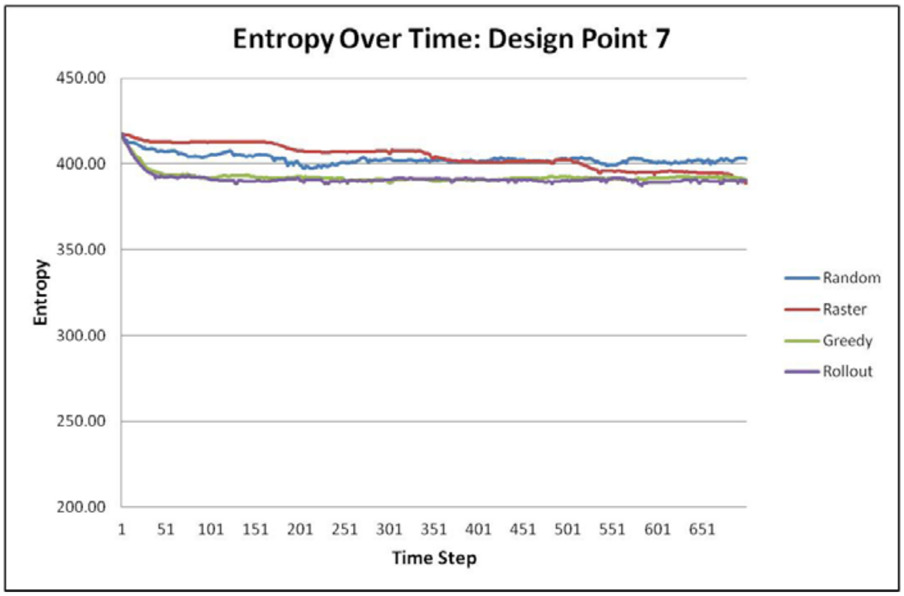


Figure 12: Entropy Over Time: Design Point 8

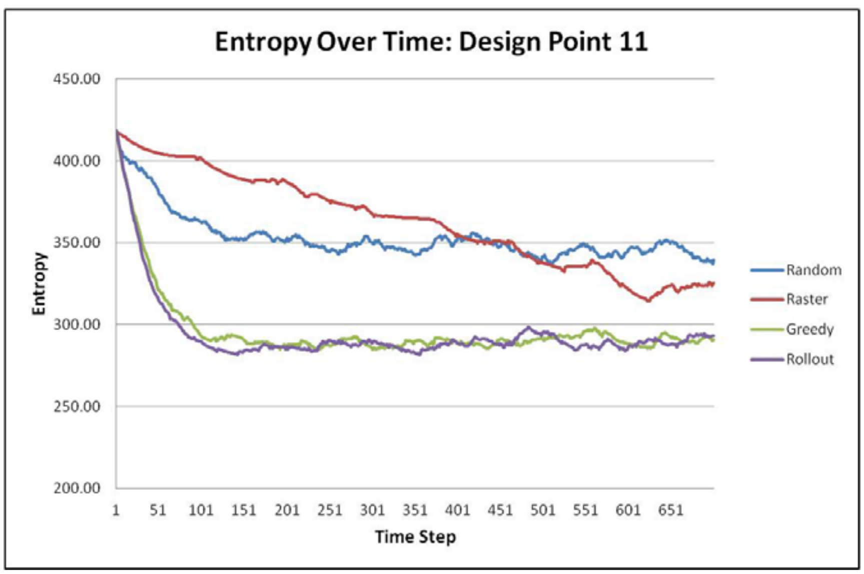


Figure 13: Entropy Over Time: Design Point 11

The rollout and greedy algorithms both produce a drastic decease in entropy at the start of the run and reach a steady state around time step 100. The raster algorithm continued to decrease over time, and did reach an entropy value similar to the greedy and rollout algorithms. However, as shown in Figure 12 and Figure 13, the entropy value

does not continue to decrease past that of the greedy or rollout algorithms. The random algorithm reached a steady state well above the other search algorithms.

Steady State Entropy Above 400

The remaining 13 design points produced runs which fell into the second group of behavior, runs which had an entropy value that did not drop below 400. Figure 14 below is the plot for the run of design point 4, and is a representative plot for all of the runs in this group.

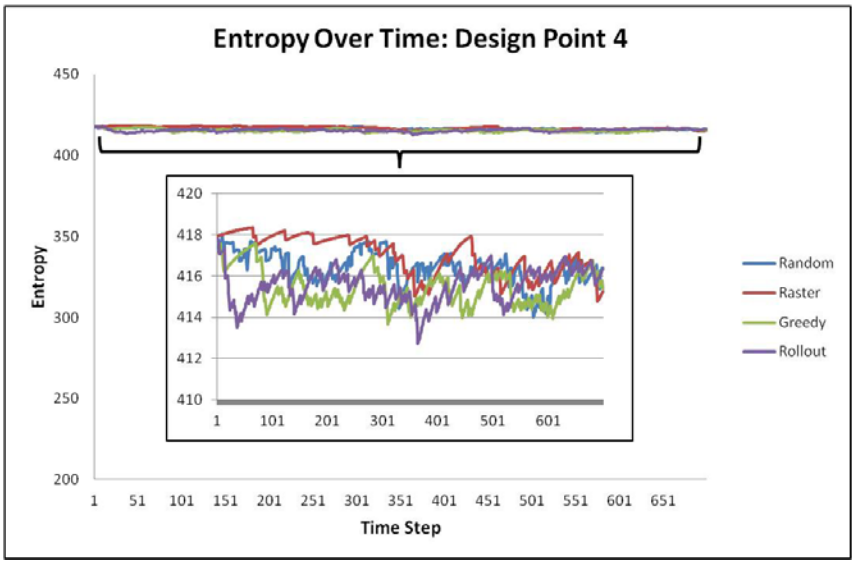


Figure 14: Entropy Over Time: Design Point 3

The rollout and greedy algorithms both produce a decrease in entropy at the start of the run greater than that of the raster and random algorithms. Although the decrease in entropy was very slight compared to Figure 12 and Figure 13 above, the greedy and rollout algorithms remained at a lower entropy value throughout the run.

Average Entropy Results

For this MOE, the average entropy was the response variable, and all of the factors from the NOLH were the predictor variables. The average entropy of the region was generated by the model for 30 replications of each design point (using Equation 15) of the NOLH over two separate scenarios: the fixed time step scenario, and the exhaustive target detection scenario. JMP 9 software was used to analyze the average entropy data for each scenario. A first order model with two way interactions was constructed using JMPs stepwise and standard least squares tools. The results for these scenarios are outlined below.

Fixed Time Step Scenario

The regression model resulting from the fixed time step runs contained all of the main effects, as well as a number of two way interactions (Table 3). The main effects were expected to have significance on the model as they were the simulation input variables that constructed the characteristics of the region. The key effect of note was the significant difference between the rollout and greedy algorithms compared to the random and raster algorithms. The estimation term signified that the greedy and rollout algorithms produced a significantly lower average entropy of the region than the raster and random algorithms. A number of two way interactions were also noteworthy.

The interaction between the number of track assets and the number of moving targets indicated a track asset's impact on the overall entropy of the region. As a track asset followed a moving target, the entropy of those hexes was continually being updated. No entropy growth took place because of that continual update, and the entropy of the region remained at a lower level. The interaction between the number of track assets and the P_{Det} indicated that with a higher P_{Det}, the more likely a track asset was designated to

track a detected target. Again, no entropy growth took place because the target location continued to be updated.

Table 3: Parameter Estimates for Fixed Time Step Scenario

Term	Estimate	p-value
Intercept	470.6954	<.0001*
# Track Assets (#TrAsset)	-3.7084	<.0001*
# Search Assets (#SrAsset)	-2.9938	<.0001*
#Stationary Targets (#StatTar)	0.267866	0.0036*
# Moving Targets (#MovTar)	0.128673	0.0002*
Probability of Detect (PDet)	-0.56185	<.0001*
Probability of Track (PTrack)	-0.21955	<.0001*
LookAhead (LookA)	1.716659	<.0001*
Algorithm(4&3-1&2)	-3.52685	<.0001*
#TrAsset*#SrAsset	12.18559	<.0001*
#TrAsset*PDet	-0.01998	<.0001*
#TrAsset*#MovTar	-0.05691	0.0028*
#TrAsset*PDet	0.968479	<.0001*
#TrAsset* Algorithm(4&3-1&2)	-1.46337	0.0053*
#SrAsset*#StatTar	1.668106	<.0001*
#SrAsset*Algorithm(4&3-1&2)	-0.99801	0.1049
#StatTar*PDet	0.073197	<.0001*
#StatTar*Algorithm(4&3-1&2)	0.136434	0.1218
#MovTar*PDet	-0.09696	<.0001*
PDet* Algorithm(4&3-1&2)	-0.16508	<.0001*
* p-value less than 0.05, indicating significance to the regression model		

Model adequacy plots on the regression were also created through JMP analysis. Figure 15a below is the plot of the residuals vs. predicted values. A visual examination of the plot indicated that the residuals do not fully follow a random pattern. This signified that there might be some non-linearity or non-constant variance patterns within the model. Transformations on the response did not rectify this issue. One possibility for this non-constant variance could be due to the initial entropy decrease in the first 100 time steps with a steady state entropy being reached for the remainder of the run. Figure

15b below is the normal probability plot of the residuals. A visual examination of the plot indicated no problem with the normality assumption of the residuals. Based on these plots, it can be concluded that the regression model may contain defects in relation to non-constant variance of the residuals.

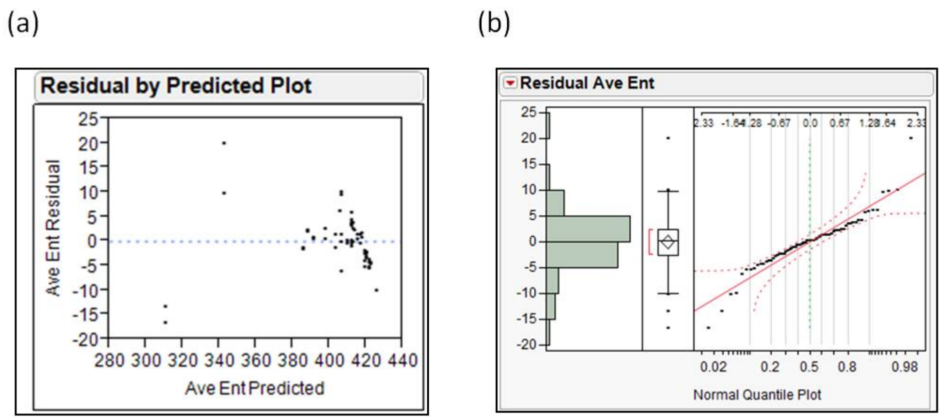


Figure 15: Model Adequacy Plots for Fixed Time Step Scenario
(a) Residual vs. Predicted Plot (b) Normal Probability Plot

Exhaustive Target Detection Scenario

The regression model resulting from the exhaustive target detection runs was very similar to the regression model from the fixed time step scenario. The model contained all of the same main effects and two way interaction terms (Table 4). Although the estimate values and p-values differed slightly, the same conclusions were drawn about the model. The key effect of note was the significant difference between the rollout and greedy algorithms compared to the random and raster algorithms. The estimation term signified that the greedy and rollout algorithms produced a significantly lower average entropy of the region then the raster and random algorithms. A number of two way interactions were also noteworthy.

These interaction terms were also similar to those in the fixed time step scenario. The interaction between the number of track assets and the number of moving targets indicated a track asset's impact on the overall entropy of the region. As a track asset followed a moving target, the entropy of those hexes was continually being updated. No entropy growth took place because of that continual update, and the entropy of the region remained at a lower level. The interaction between the number of track assets and the P_{Det} indicated that with a higher P_{Det}, the more likely a track asset was designated to track a detected target. Again, no entropy growth took place because the target location continued to be updated.

Table 4: Parameter Estimates for Exhaustive Target Detection Scenario

Term	Estimate	p-value
Intercept	445.9396	<.0001*
# Track Assets (#TrAsset)	-3.81058	<.0001*
# Search Assets (#SrAsset)	-2.49985	<.0001*
#Stationary Targets (#StatTar)	0.211112	0.0045*
# Moving Targets (#MovTar)	0.471338	<.0001*
Probability of Detect (PDet)	-0.51543	<.0001*
Probability of Track (PTrack)	-0.12521	<.0001*
LookAhead (LookA)	2.697389	<.0001*
Algorithm(4&3-1&2)	-2.54101	0.0001*
#TrAsset*#TrAsset	18.44641	<.0001*
#TrAsset*#StatTar	3.08914	<.0001*
#TrAsset*#MovTar	0.401585	0.0014*
#TrAsset*PDet	0.960587	<.0001*
#TrAsset*LookA	17.61195	<.0001*
#TrAsset*Algorithm(4&3-1&2)	-1.13821	0.0060*
#SrAsset*PTrack	0.479182	<.0001*
#SrAsset*LookA	-0.73872	0.1279
#SrAsset*Algorithm(4&3-1&2)	0.118777	0.0889
#StatTar*Algorithm(4&3-1&2)	-0.08584	<.0001*
PDet*PDet	445.9396	<.0001*
* p-value less than 0.05, indicating significance to the regression model		

Model adequacy plots on this regression were also created through JMP analysis. Again, these plots were very similar to the model adequacy plots from the fixed time step scenario. Figure 16a below is the plot of the residuals vs. predicted values. A visual examination of the plot indicated that the residuals do not fully follow a random pattern. This signified that there might be some non-linearity or non-constant variance patterns within the model. Transformations on the response did not rectify this issue. One possibility for this non-constant variance could be due to the initial entropy decrease in the first 100 time steps with a steady state entropy being reached for the remainder of the run. Figure 16b below is the normal probability plot of the residuals. A visual examination of the plot indicated no problem with the normality assumption of the residuals. Based on these plots, it can be concluded that the regression model may contain defects in relation to non-constant variance of the residuals.

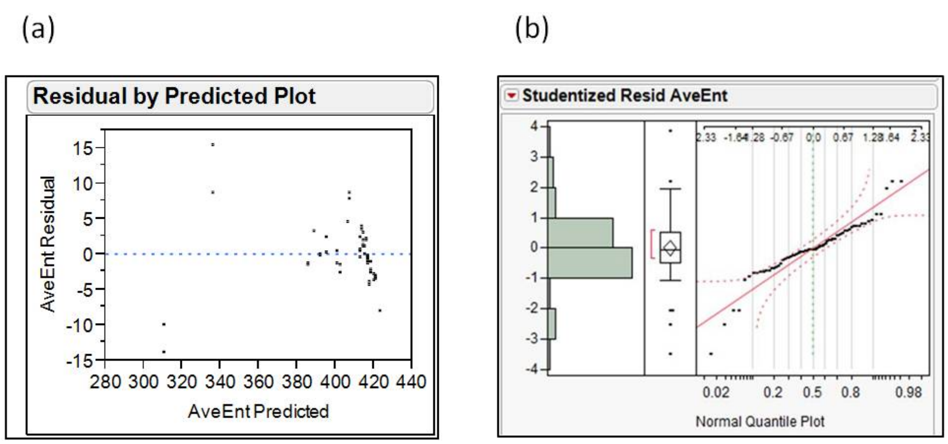


Figure 16: Model Adequacy Plots for Exhaustive Target Detection Scenario (a) Residual vs. Predicted Plot (b) Normal Probability Plot

Number of Targets Found Results

For this MOE, the number of targets found was the response variable, and all of the factors from the NOLH were the predictor variables. The number of targets found was the MOE specific to the fixed time step scenario. Data was generated by the model for 30 replications of each design point of the NOLH. JMP 9 software was used to analyze the results, and a first order model with two way interactions was constructed using stepwise and standard least squares tools. The resulting regression model contained all of the main effects, as well as a number of two way interactions (Table 5). The key effect of note was the significant difference between the rollout and greedy algorithms compared to the random and raster algorithms. The estimation term signified that the greedy and rollout algorithms produced a higher number of targets found than the raster and random algorithms. Also of note was the significant difference between the random and raster algorithms. The estimation term signified that the raster algorithm produced a higher number of targets found than the random algorithm. A number of two way interactions were also noteworthy.

The interaction between the number of track assets and the probability of detect indicated that with a higher probability of detect, the more likely a track asset was designated to track a detected target. As a track asset is following a moving target, the entropy of those hexes was continually updated. No entropy growth took place because the target location continued to be updated. The interaction between the number of stationary and moving targets indicated that as the amount of targets increases, the more likely a higher number of targets would be found within a set number of time steps.

Another noteworthy interaction is between the parameter estimates of the look ahead main effect, and the look ahead quadratic effect. The positive parameter estimate

of the look ahead main effect indicated that as the look ahead number increased, the number of targets found increased as well. However, the negative parameter estimate associated with the look ahead quadratic term indicated that as the look ahead value increased, it decreased the number of targets found. These competing estimates indicated that there was a point at which the look ahead value changed between adding to the number of targets found and detracting from the number of targets found. Based on the parameter estimates and the range of look aheads varying from two through eight, the number of look aheads that created the most positive value to the number of targets found was a look ahead of three.

Table 5: Parameter Estimates for Number of Targets Found

Term	Estimate	p-value
Intercept	-14.8299	<.0001*
# Track Assets (#TrAsset)	1.871735	<.0001*
# Search Assets (#SrAsset)	2.16504	<.0001*
#Stationary Targets (#StatTar)	0.674777	<.0001*
# Moving Targets (#MovTar)	0.647688	<.0001*
Probability of Detect (PDet)	0.144877	<.0001*
Probability of Track (PTrack)	-0.06155	<.0001*
Look Ahead (LookA)	2.99346	0.0003*
Algorithm (2&1 – 3&4)	-2.08043	<.0001*
Algorithm (2– 1)	-1.33471	<.0001*
#TrAsset*#SrAsset	-2.06733	<.0001*
#TrAsset*PDet	0.450506	0.0003*
#SrAsset*Algorithm(2-1)	-0.26991	0.095
#StatTar*#MovTar	-.045654	0.0293*
#StatTar*PTrack	-0.00992	<.0001*
#StatTar*Algorithm(2&1-3&4)	-0.03161	0.0554
#MovTar*#PDet	0.008138	0.0045*
#MovTar* Algorithm(2&1-3&4)	-0.08088	<.0001*
#MovTar* Algorithm(2-1)	-0.08442	0.0006*
PDet* Algorithm(2&1-3&4)	-0.01581	0.0094*
PDet* Algorithm(2-1)	0.017589	0.0382*
PTrack* Algorithm(2&1-3&4)	-0.00869	0.1425
PTrack* Algorithm(2-1)	-0.0121	0.1489
LookA*LookA	-0.50506	0.0010*
* p-value less than 0.05, indicating significance to the regression model		

Model adequacy plots on the regression were also created through JMP analysis. Figure 17a below is the plot of the residuals vs. predicted values. A visual examination of the plot indicated no pattern to the residuals. This signified that there was no non-linearity or non-constant variance patterns within the model. Figure 17b below is the normal probability plot of the residuals. A visual examination of the plot indicated no problem with the normality assumption of the residuals. Based on these plots, it can be concluded that the regression model does not contain any adequacy defects.

(a) (b)

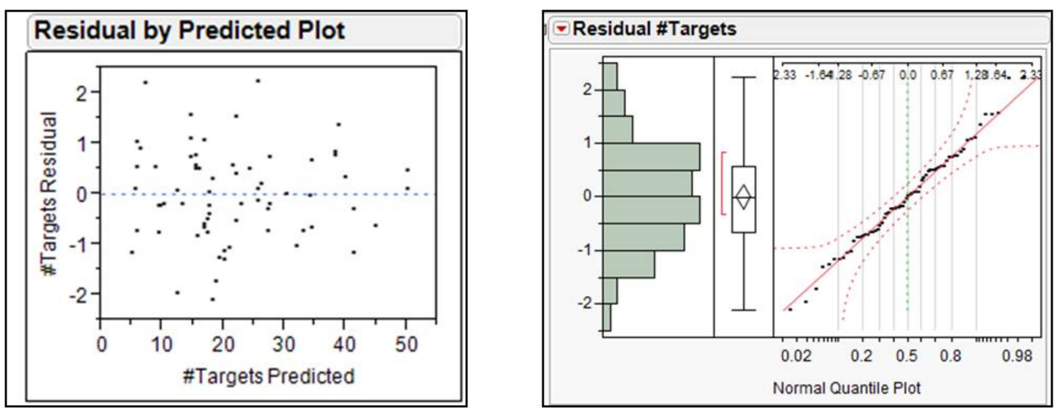


Figure 17: Model Adequacy Plots for Number of Targets Found
(a) Residual vs. Predicted Plot (b) Normal Probability Plot

Time Steps to Completion Results

For this MOE, the number of time steps to completion was the response variable, and all of the factors from the NOLH were the predictor variables. The number of time steps to completion was the MOE specific to the exhaustive target detection scenario. Data was generated by the model for 30 replications of each design point of the NOLH. JMP 9 software was used to analyze the results and a first order model with two way interactions was constructed using stepwise and standard least squares tools. The

resulting regression model contained all of the main effects, as well as a number of two way interactions (Table 6). The key effect of note was the significant difference between the rollout, greedy, and raster algorithms compared to the random algorithm. The estimation term signified that the rollout, greedy and raster algorithms found all of the targets within the region in less time steps than the random algorithm. Also of note was the significant difference between the rollout algorithm and the greedy and raster algorithms. The estimation term signified that the rollout algorithm found all of the targets within the region in less time steps than the greedy and raster algorithm. A number of two way interactions were also noteworthy.

The interaction between the number of track assets and search assets indicated that the total number of assets had an effect of the number of time steps to completion. The higher the number of assets, the fewer time steps were needed to find all the targets within the region. The interaction between the number of track assets and the P_{Track} indicated that the ability to find new targets was impacted by the number of track assets and the amount of time those track assets were locked on a target. The greater the number of track assets and higher the P_{Track} lead to a greater amount of time steps to completion.

Table 6: Parameter Estimates for Time Steps to Completion

Term	Estimate	p-value
Intercept	4933.421	<.0001*
# Track Assets (#TrAsset)	-357.349	<.0001*
# Search Assets (#SrAsset)	-562.47	<.0001*
#Stationary Targets (#StatTar)	54.31889	<.0001*
# Moving Targets (#MovTar)	83.04373	<.0001*
Probability of Detect (PDet)	-41.4556	<.0001*
Probability of Track (PTrack)	-13.8377	<.0001*
LookAhead (LookA)	186.9952	<.0001*
Algorithm(2&3&4-1)	-376.331	<.0001*
Algorithm(4-2&3)	-43.0721	0.0196*
#TrAsset*#TrAsset	619.1614	<.0001*
#TrAsset*#SrAsset	3337.397	<.0001*
#TrAsset*PDet	-5.39326	0.0653
#TrAsset*PTrack	-23.8687	<.0001*
#TrAsset*Algorithm(2&3&4-1)	152.7244	<.0001*
#SrAsset*#SrAsset	-953.283	<.0001*
#SrAsset*#MovTar	122.7542	<.0001*
#SrAsset*Algorithm(2&3&4-1)	142.1504	0.0014*
#StatTar*Algorithm(2&3&4-1)	-17.6953	0.0056*
#StatTar*Algorithm(4-2&3)	-11.0676	0.0877
PDet*PTrack	-10.3367	<.0001*
PDet*Algorithm(4-2&3)	5.468495	0.44750
LookA*Algorithm(2&3&4-1)	-37.9126	0.1830
* p-value less than 0.05, indicating significance to the regression model		

Model adequacy plots on the regression were also created through JMP analysis. Figure 18a below is the plot of the residuals vs. predicted values. A visual examination of the plot indicated no pattern to the residuals. This signified that there was no non-linearity or non-constant variance patterns within the model. Figure 18b below is the normal probability plot of the residuals. A visual examination of the plot indicated there was no problem with the normality assumption of the residuals. Based on these plots, it can be concluded that the regression model does not contain any adequacy defects.

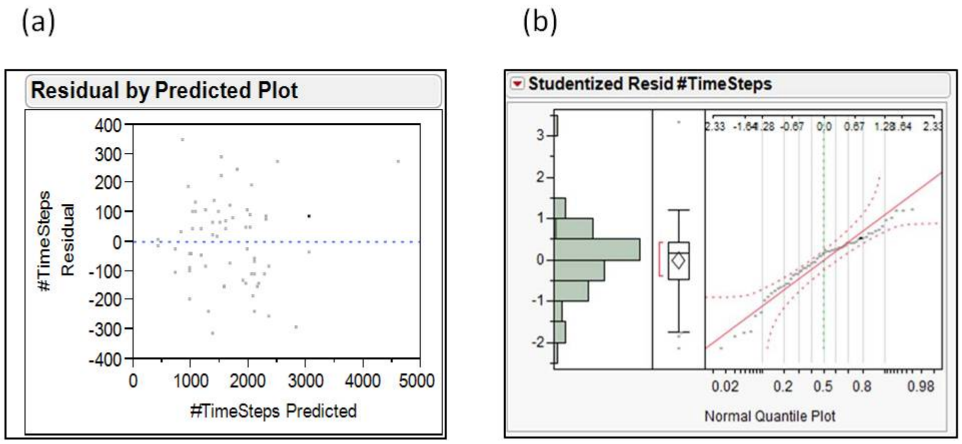


Figure 18: Model Adequacy Plots for Time Steps to Completion (a) Residual vs. Predicted Plot (b) Normal Probability Plot

Surveillance versus Reconnaissance

The step entropy was also used to analyze the balance of allocating search assets, used primarily for surveillance, and track assets, used to track identified targets. This analysis was performed by holding all of the model input values constant, except for the allocation of search and track assets, P_{Det}, and P_{Track}. The total number of assets was held at four for this analysis. The allocation of search assets varied from one to four, and the allocation of track assets varied from zero to three. The step entropy was generated by the model for each combination of assets, over various values of P_{Det} and P_{Track}. The step entropy was plotted over time for each run to provide insights on the balance of asset allocation.

Two main trends resulted from this analysis. The first trend involved values of P_{Det}, and P_{Track} greater than 0.95. As the number of search assets increased, the steady state entropy of the region decreased respectively. Figure 19 and Figure 20 below are a visual representation of this trend. In Figure 19, the steady state entropy value with one

search asset and three track assets was approximately 370. This value decreased to approximately 355 as the allocated search assets reached four. The same trend holds true in Figure 20, although the decrease in the entropy is not as drastic. As the P_{Det}, and P_{Track} decrease, the second trend becomes apparent: a decrease in variability in the steady state entropy value.

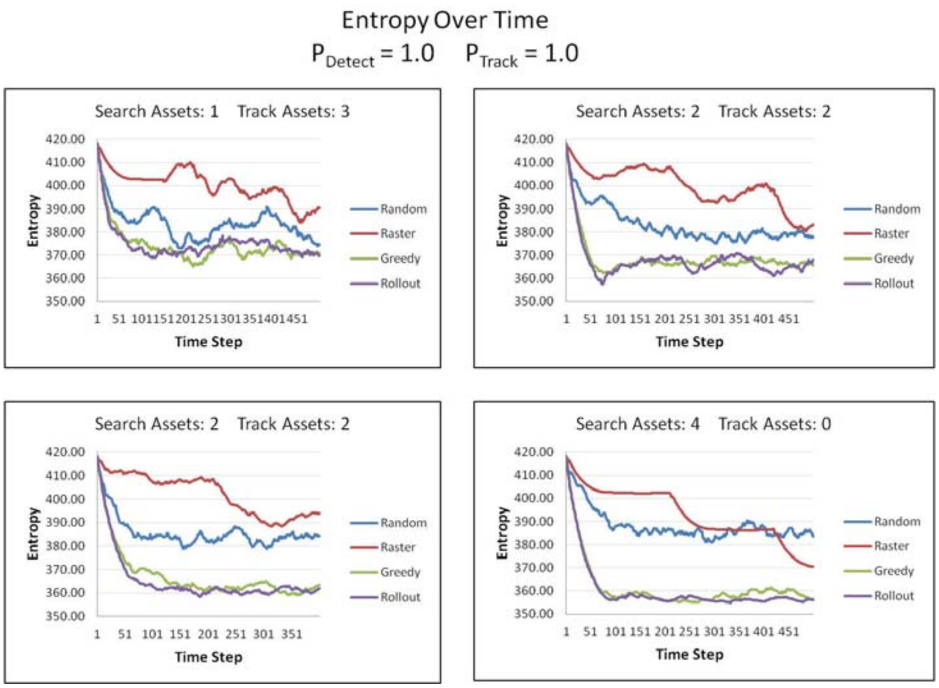


Figure 19: Entropy Over Time - PDet = 1.0, PTrack = 1.0

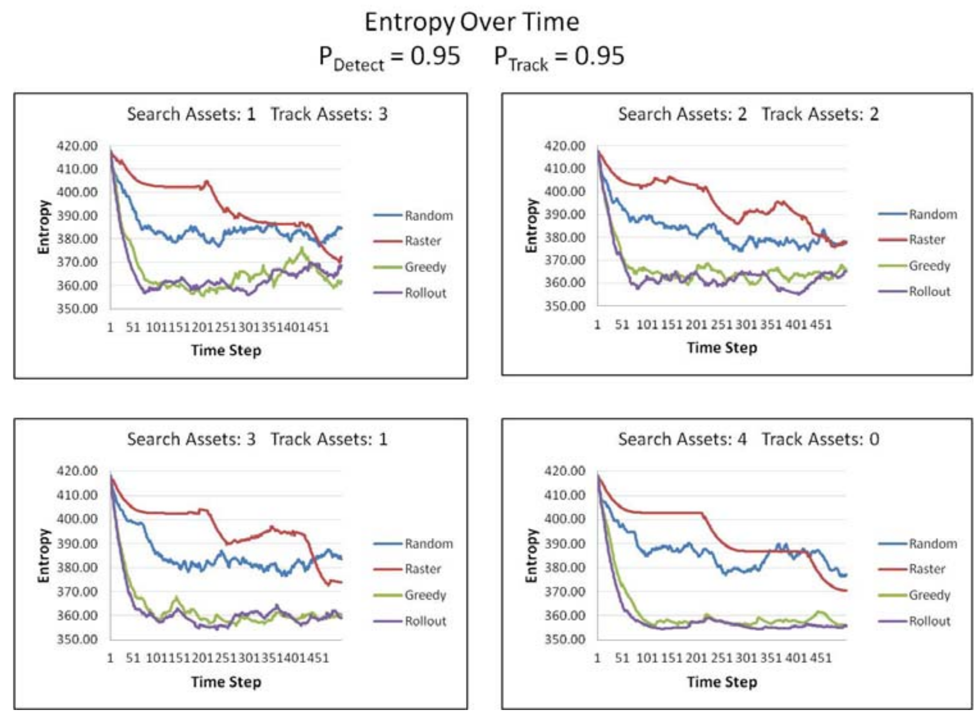


Figure 20: Entropy Over Time - PDet = 0.95, PTrack = 0.95

Although the steady state entropy did not decrease drastically for P_{Det}, and P_{Track} less than 0.95, Figure 21 and Figure 22 are a visual representation of how the steady state is affected by the allocation of search and track assets for lower probabilities of detect and track. As the number of search assets increased, the steady state entropy of the region creates a less variable steady state. The fluctuation of the steady state entropy value caused by the track assets decreased as the number of track assets decreased.

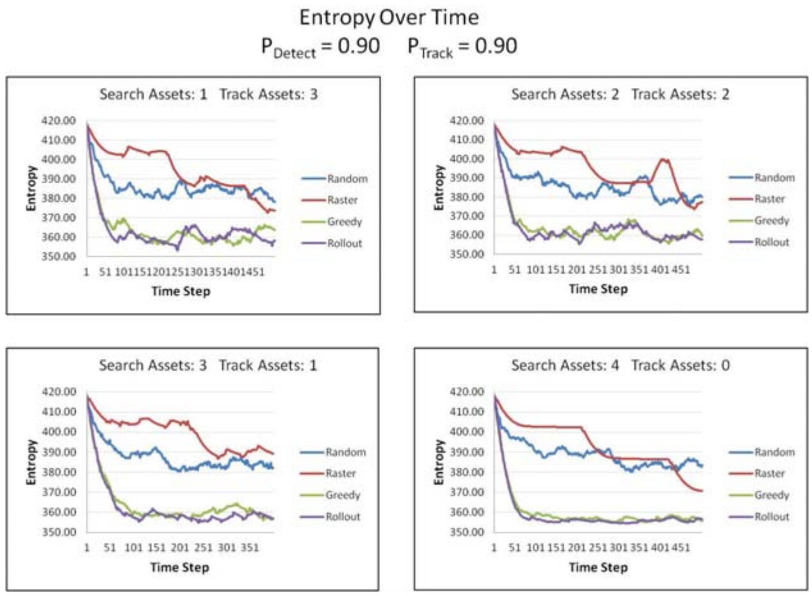


Figure 21: Entropy Over Time - PDet = 0.90, PTrack = 0.90

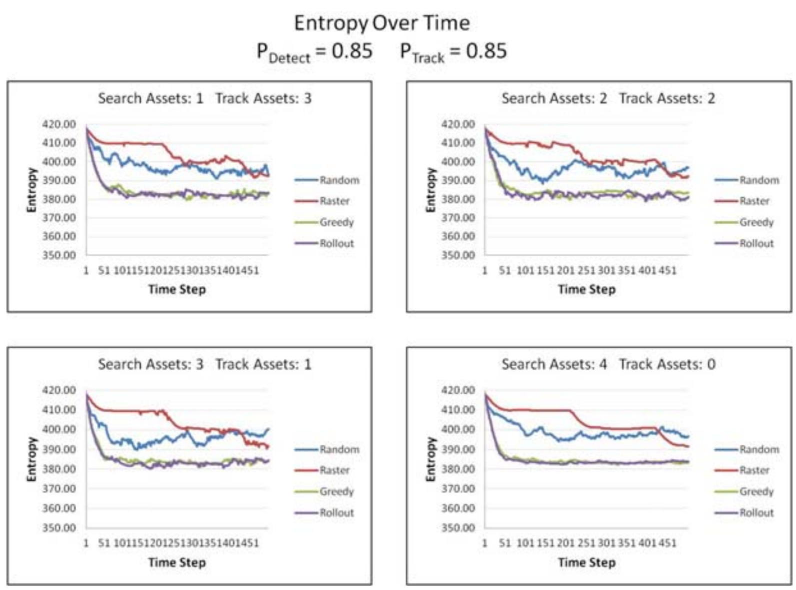


Figure 22: Entropy Over Time - PDet = 0.85, PTrack = 0.85

Conclusions

Based on these initial results, the rollout search algorithm could provide military decision makers with an effective way to utilize ISR assets to minimize the amount of uncertainty within a region. In addition to minimizing uncertainty in the first 100 time steps of the run, the rollout algorithm also produced the highest number of targets found within the fixed time step scenario, and, for the exhaustive target detection scenario, discovered all of the targets within the region in less time steps. Based on these results, the rollout algorithm provides superior performance in the allocation of ISR assets while balancing detection of new targets versus surveillance of already detected targets.

Additionally, trends into the allocation of assets allow for decision makers to efficiently balance the number of search and track assets within a region. Search assets can be utilized to create a smoother steady state entropy value for the region, compared to track assets alone. Furthermore, given a fixed number of ISR assets, increasing the number of tracking assets adds to the variability of the steady state entropy. Therefore, track assets should be allocated when only targets of high interest need to be tracked.

V. Conclusions and Recommendations

This chapter is organized into two main sections: recommendations based on analysis, and future research.

Recommendations

This research sought to use entropy as a measure of effectiveness to balance detection of new targets versus surveillance of already detected targets within a bounded domain. Based on the analysis in the previous chapter, the rollout algorithm provides superior performance in the allocation of ISR assets, thereby providing military decision makers with an effective way to minimize the amount of uncertainty within a region. Furthermore, search assets can be utilized to drive down the steady state entropy of a region, as well as create a smoother steady state entropy value for the region. Based on these results and analysis, there are a number of recommendations for future research in this area.

Future Research

Results Based Research

Based on the average entropy analysis, the residual vs. predicted plot indicated that the regression model constructed might have a non-constant variance. One possibility for this non-constant variance could be due to the initial entropy decrease in the first 100 time steps with a steady state entropy being reached for the remainder of the run. A more thorough examination of this data may provide further insight into this and

determine whether the model is appropriate to use for this MOE, or whether a different MOE is more suitable.

Based on the results from the step entropy MOE, the first 100 time steps are critical to the decrease in the entropy of the region. Once a steady state has been reached, an analysis should be performed on switching search algorithms. This might produce a methodology to drive the entropy down to an even lower steady state entropy value.

Another area of future research would be a more in depth look at the look ahead values associated with the rollout algorithm. The relationship between the look ahead main and quadratic terms was only found in one of the MOEs, but may have an effect on the others if further explored. Also, entropy growth was not considered in the "cost to go" calculation, but could have an effect on the decision that is made. Therefore, both the entropy growth rate, and the number of look aheads should be explored further.

Assumption Based Research

A number of assumptions were made for the baseline model. These assumptions should be addressed and the model expanded to incorporate a greater scope within the model design. These changes include, but are not limited to:

(1) *Heterogeneous Asset Force.* Commanders are not always limited to one type of ISR asset or sensor. The ability to use more than one type of ISR asset or sensor should be incorporated within the model. This will allow for the possibility of wide and narrow search options, including cueing between them. The model would also need to incorporate the possibility of having more than one target within the sensor view.

(2) *Ground Truth and Diffusion Models*. Ground truth, such as road, buildings, and terrain, should be incorporated within the model to create a more realistic environment. This will restrict target movements and expand the types of diffusion models to be used on those identified targets within the model.

(3) *False Alarm Rate*. False alarms are a reality in target detection. False alarm rates should be incorporated within the model. This will increase the uncertainty surrounding target detection, but is a more realistic sensor model.

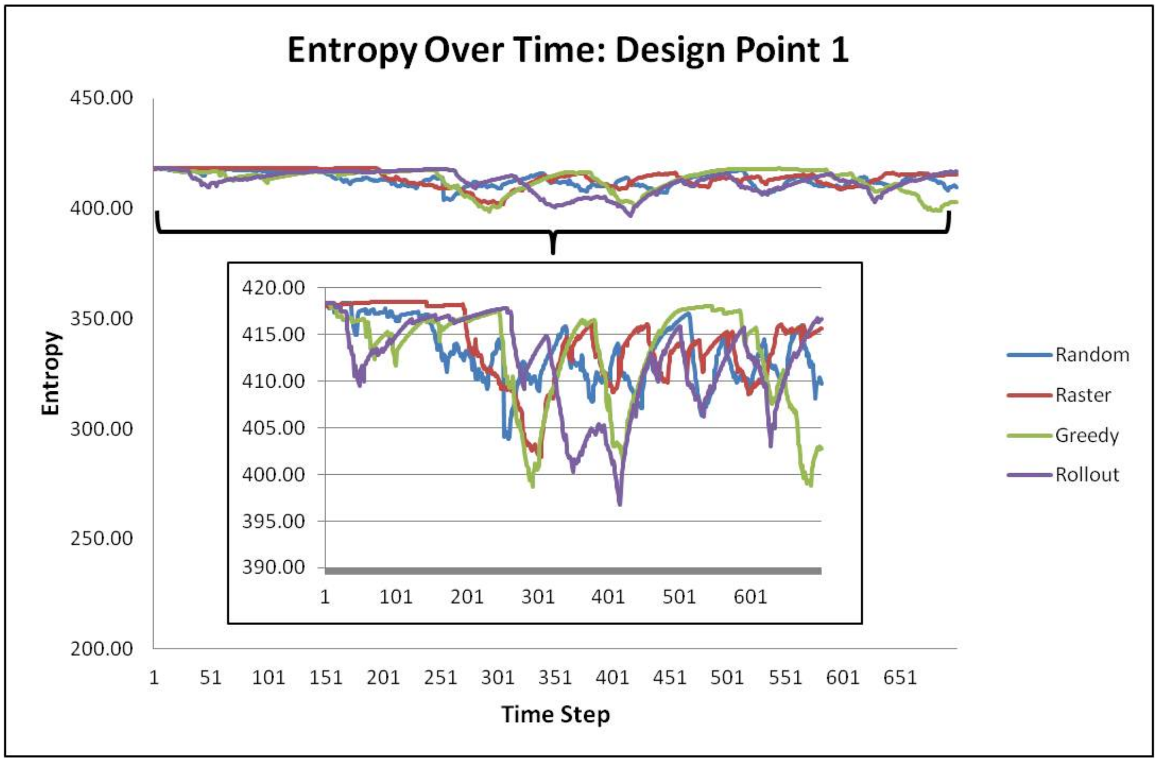
(4) *Non-unique target identifiers*. Targets are not always distinguishable from each other. A certain number of targets may be uniquely identifiable, but others should be identified by target type instead of separate identifiers. This will increase the uncertainty surrounding the targets within the region, but is a more realistic environment.

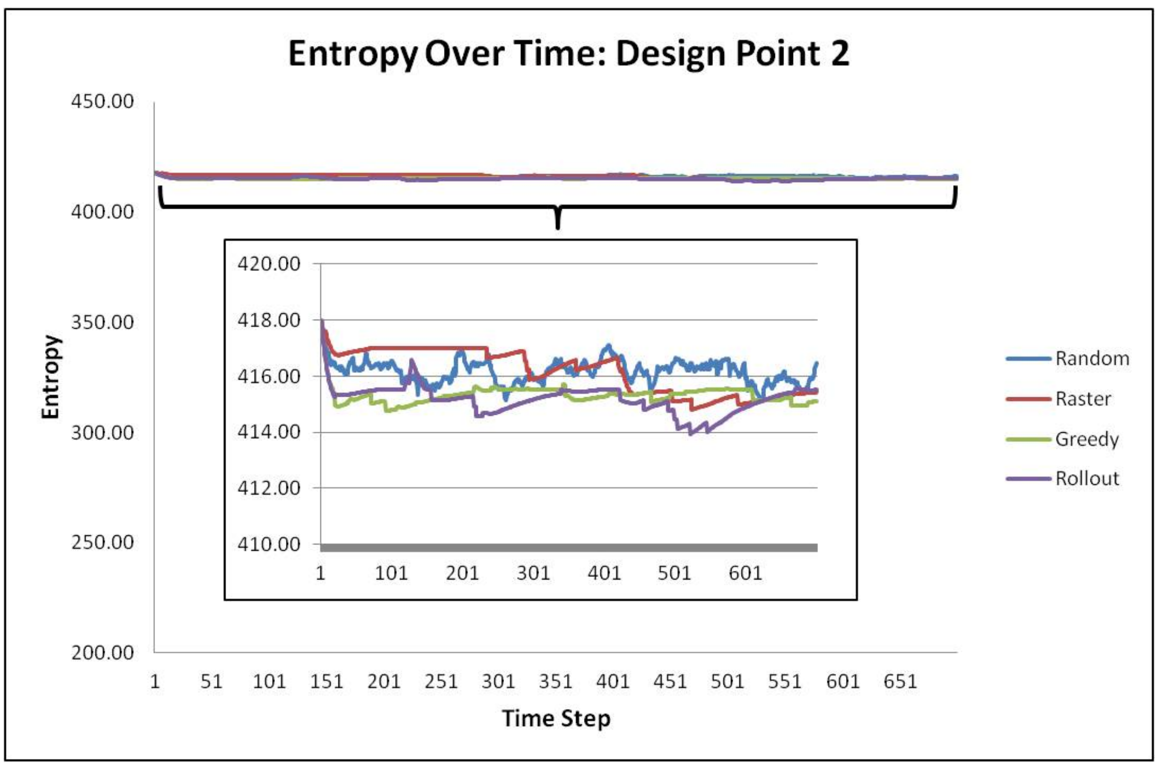
(5) *Targets of varying importance*. Not all targets are of equal value to a commander. As targets are identified, a level of importance should be applied to that target. A track asset would then be assigned to track targets with certain levels of importance to the commander.

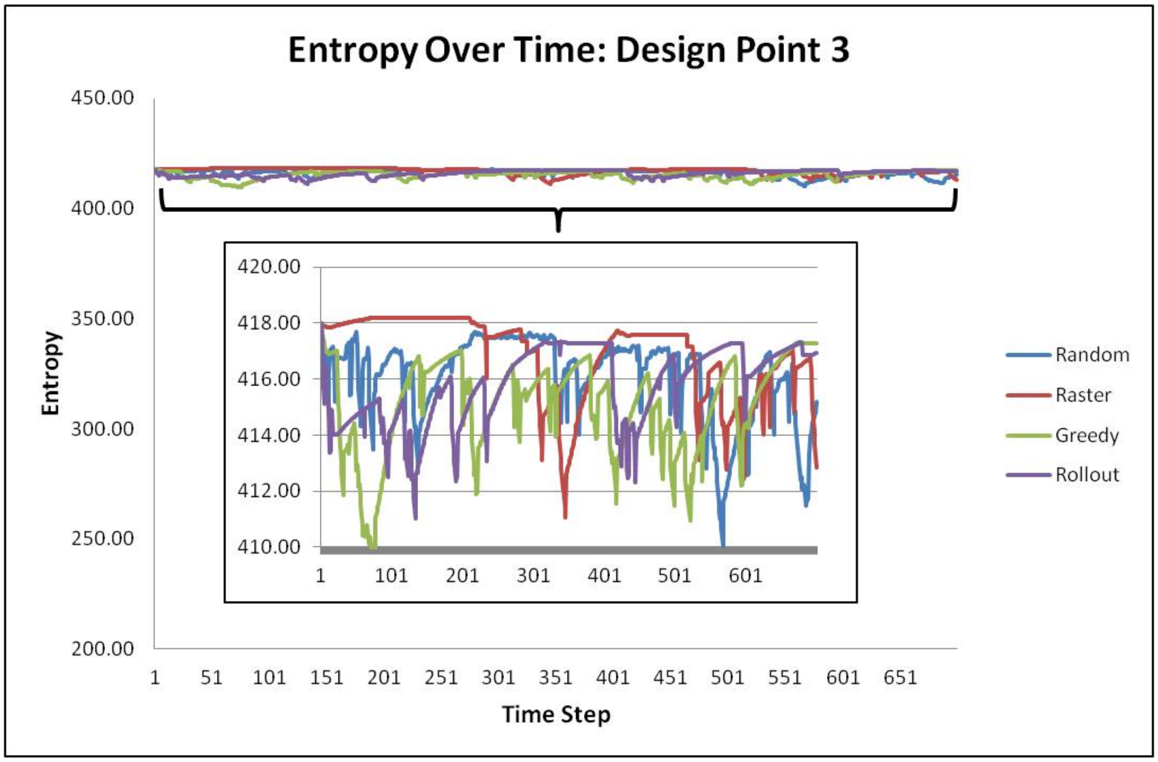
Appendix A

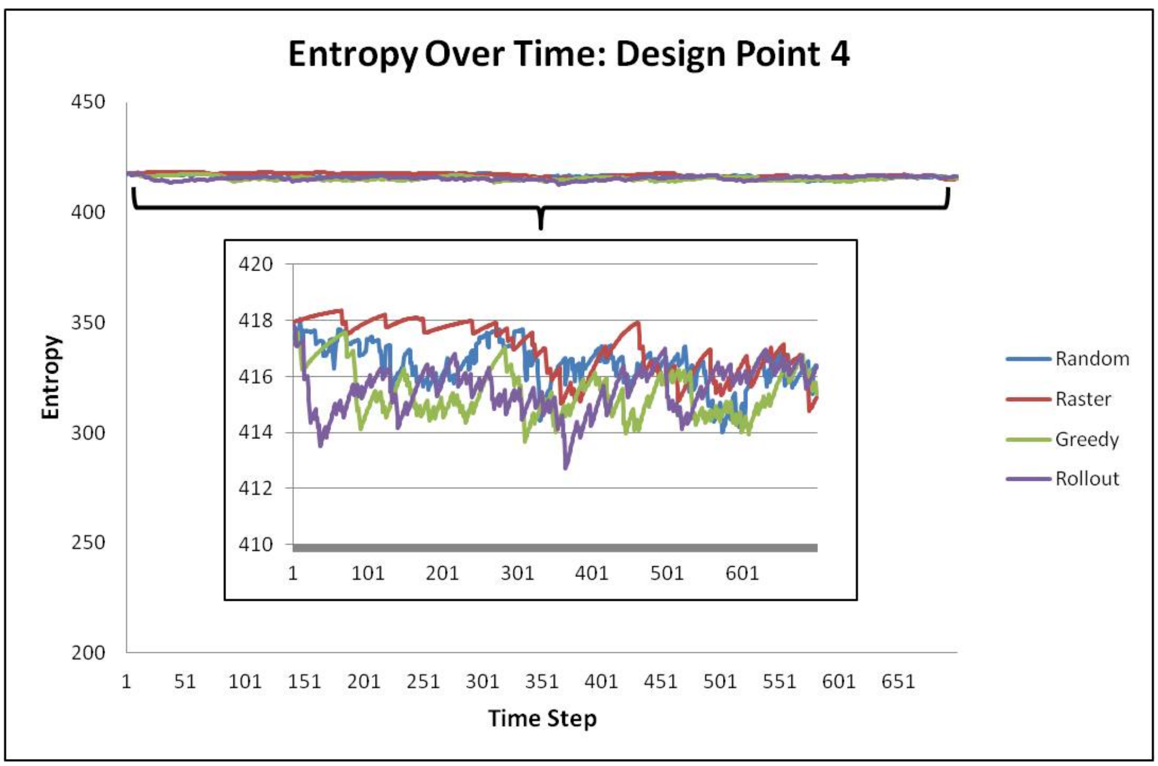
This appendix includes tables outlining all of the NOLH design points, as well as their corresponding Entropy Over Time plots for the Step Entropy MOE.

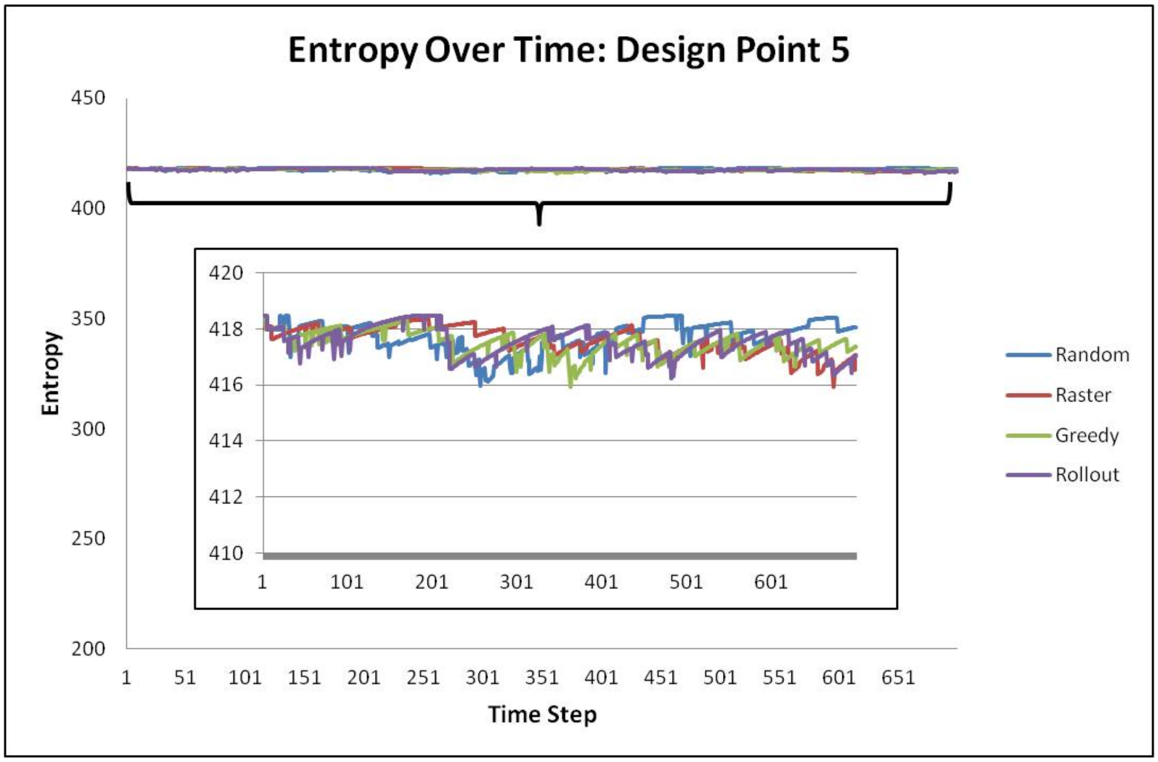
Factor	# Search Assets	# Track Assets	Probability Detect	Probability Track	# Stationary Targets	# Moving Targets	# Look Aheads
Low Level	1	0	0.20	0.20	1	1	2
High Level	5	5	1.0	1.0	30	30	8
Design Points	**Design Point Levels**						
1	5	2	0.40	0.95	25	12	5
2	2	0	0.20	0.45	26	17	6
3	3	1	0.70	0.85	3	8	8
4	4	1	0.65	0.30	10	30	7
5	5	4	0.45	0.20	14	5	7
6	2	5	0.25	0.80	12	25	7
7	2	3	0.90	0.55	30	10	8
8	5	3	0.85	0.70	23	28	6
9	3	3	0.60	0.60	16	16	5
10	1	3	0.80	0.25	6	19	5
11	4	5	1.00	0.75	5	14	4
12	3	4	0.50	0.35	28	23	2
13	3	4	0.55	0.90	21	1	4
14	1	1	0.75	1.00	17	26	3
15	4	0	0.95	0.40	19	6	3
16	4	2	0.30	0.65	1	21	2
17	2	2	0.35	0.50	8	3	4


Entropy Over Time: Design Point 1
Entropy
Time Step
Random
Raster
Greedy
Rollout
450.00
400.00
350.00
300.00
250.00
200.00
1
51
101
151
201
251
301
351
401
451
501
551
601
651
420.00
415.00
410.00
405.00
400.00
395.00
390.00

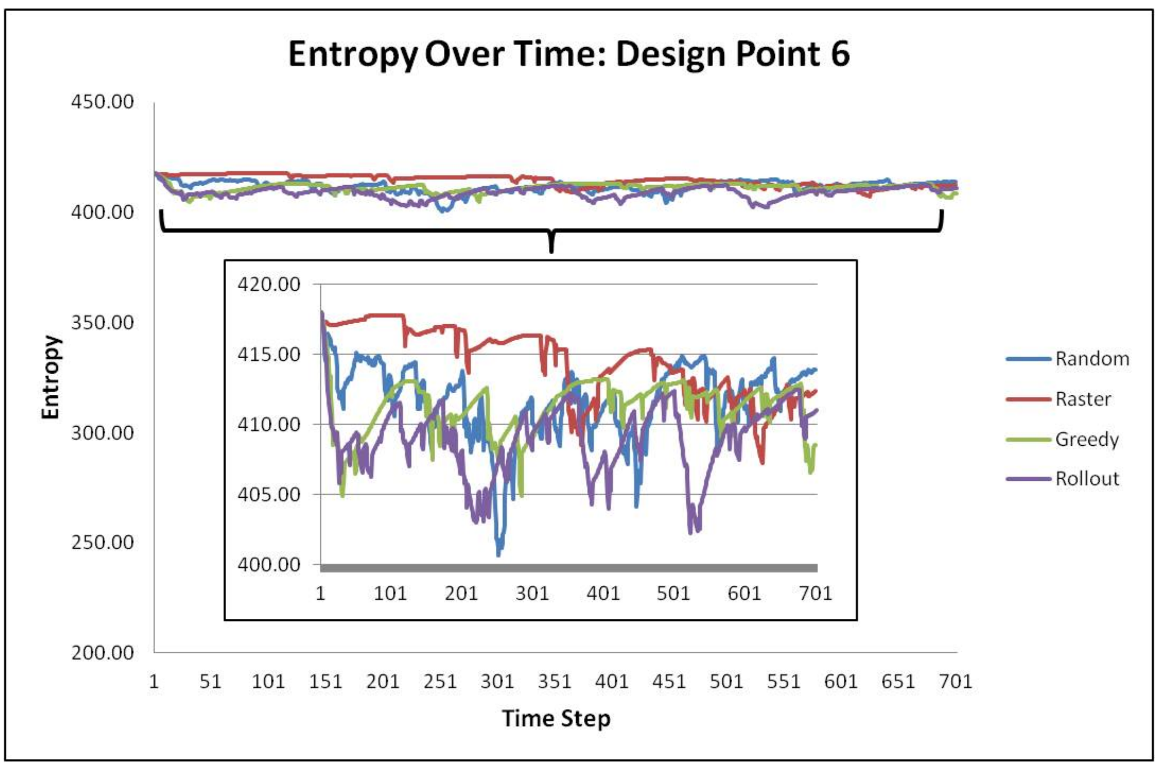

Entropy Over Time: Design Point 2
Entropy
Time Step
Random
Raster
Greedy
Rollout
450.00
400.00
350.00
300.00
250.00
200.00
1
51
101
151
201
251
301
351
401
451
501
551
601
651
420.00
418.00
416.00
414.00
412.00
410.00


Entropy Over Time: Design Point 3
450.00
400.00
350.00
300.00
250.00
200.00
Entropy
420.00
418.00
416.00
414.00
412.00
410.00
1
101
201
301
401
501
601
1
51
101
151
201
251
301
351
401
451
501
551
601
651
Time Step
Random
Raster
Greedy
Rollout

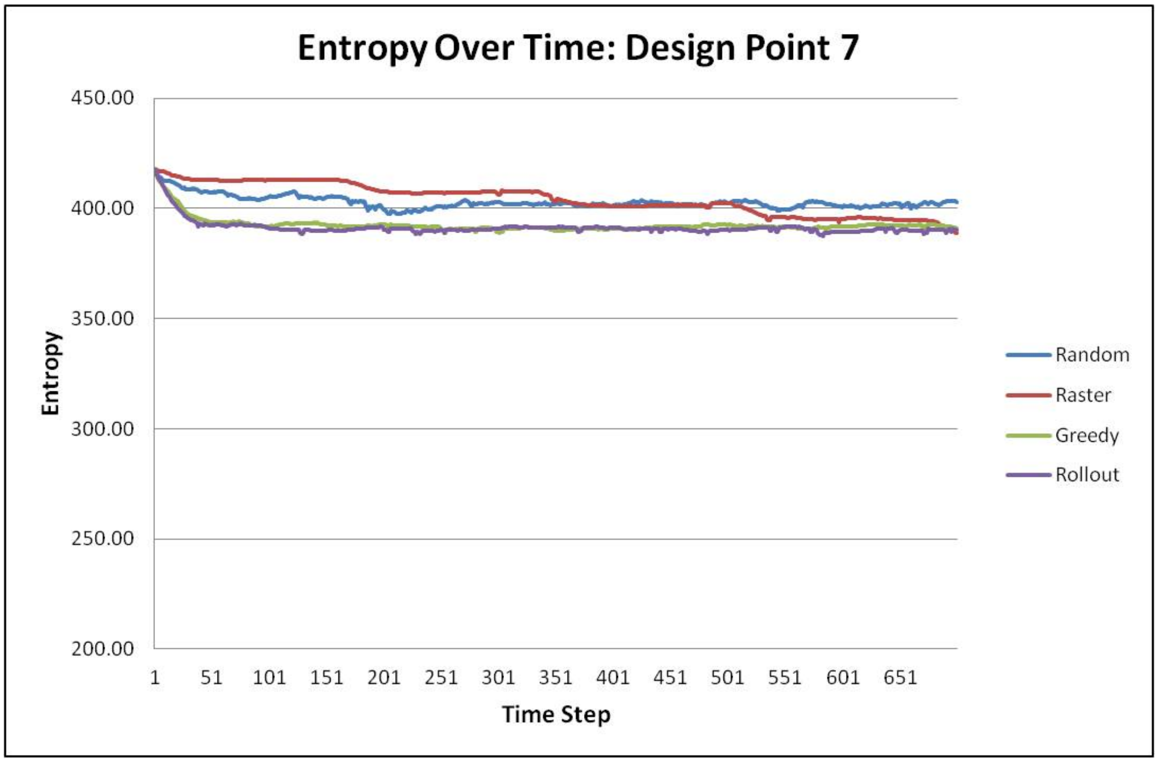

Entropy Over Time: Design Point 4
450
400
350
300
250
200
Entropy
420
418
416
414
412
410
1
101
201
301
401
501
601
1
51
101
151
201
251
301
351
401
451
501
551
601
651
Time Step
Random
Raster
Greedy
Rollout

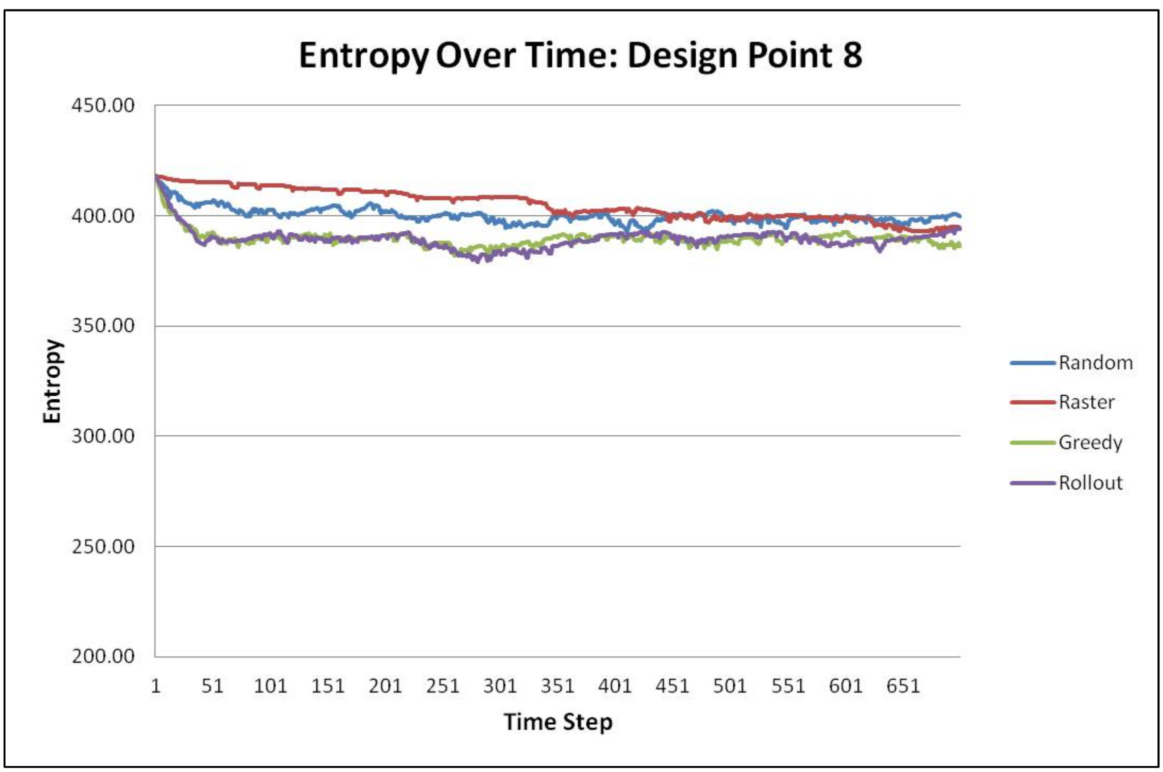


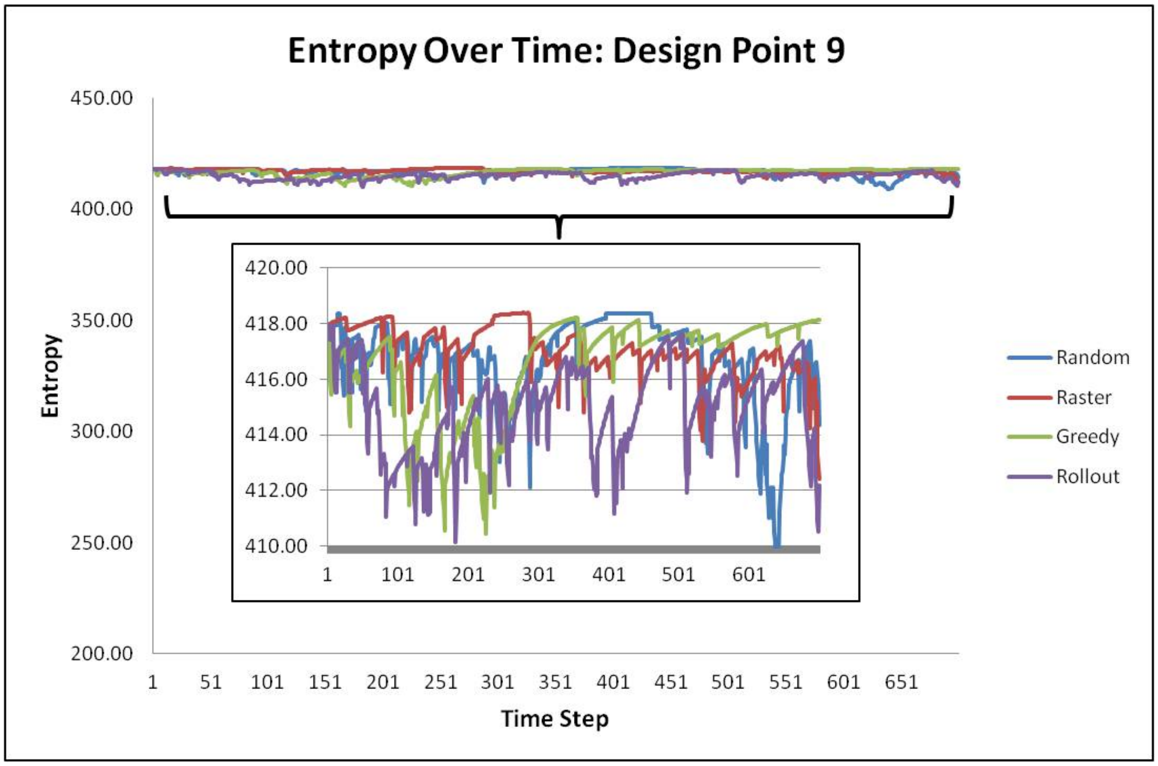
Entropy Over Time: Design Point 5
Entropy
Time Step
450
400
350
300
250
200
1
51
101
151
201
251
301
351
401
451
501
551
601
651
420
418
416
414
412
410
1
101
201
301
401
501
601
Random
Raster
Greedy
Rollout



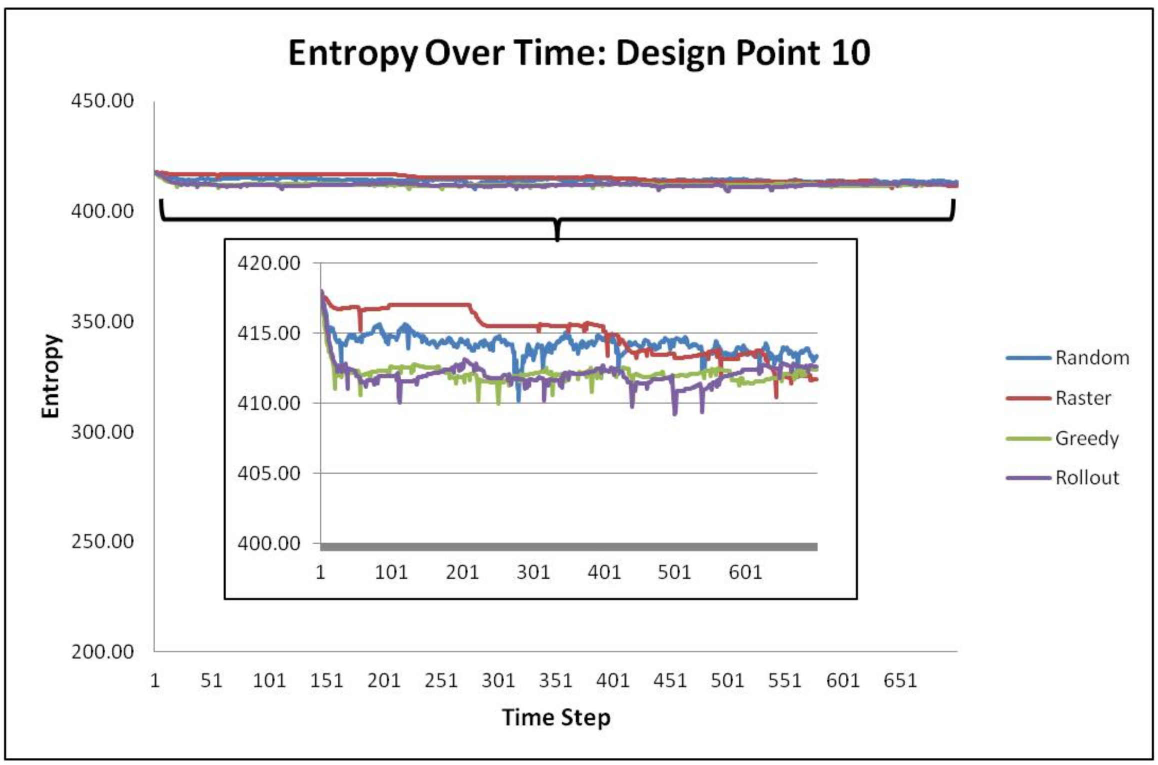
Entropy Over Time: Design Point 6
Entropy
Time Step
450.00
400.00
350.00
300.00
250.00
200.00
1
51
101
151
201
251
301
351
401
451
501
551
601
651
701
420.00
415.00
410.00
405.00
400.00
1
101
201
301
401
501
601
701
Random
Raster
Greedy
Rollout


Entropy Over Time: Design Point 7
450.00
400.00
350.00
300.00
250.00
200.00
Entropy
1
51
101
151
201
251
301
351
401
451
501
551
601
651
Time Step
Random
Raster
Greedy
Rollout

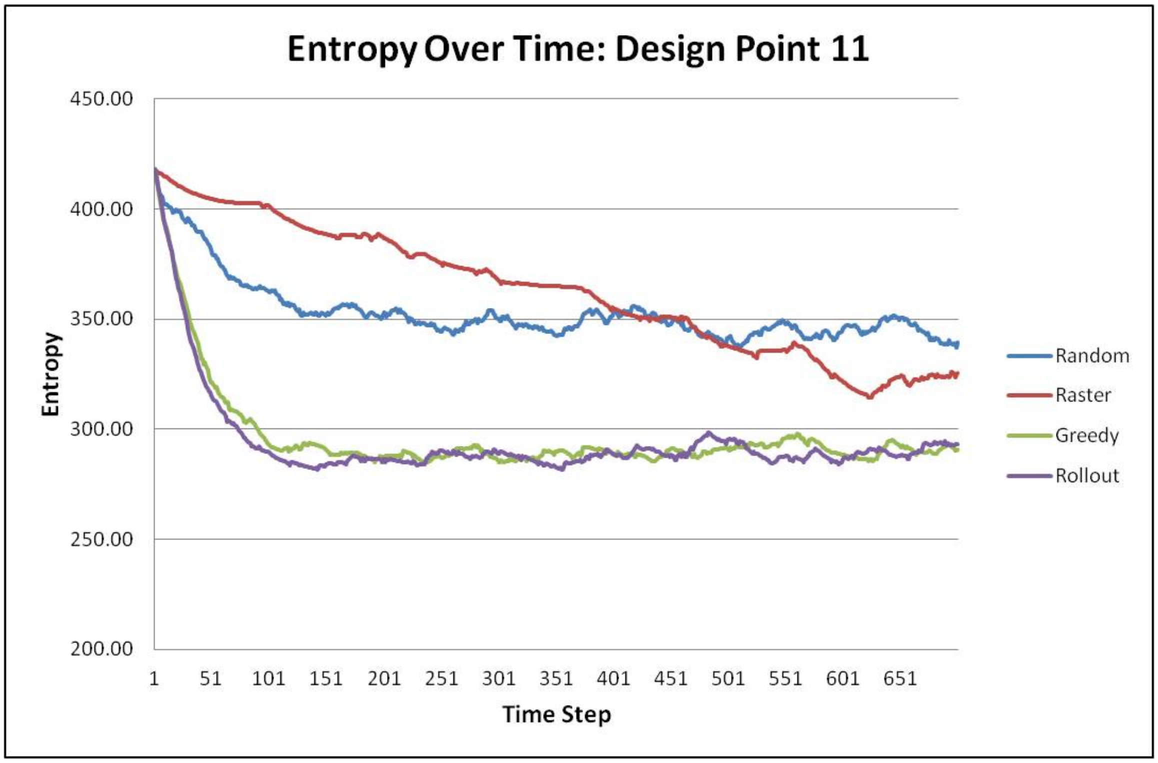

Entropy Over Time: Design Point 8
450.00
400.00
350.00
300.00
250.00
200.00
Entropy
1
51
101
151
201
251
301
351
401
451
501
551
601
651
Time Step
Random
Raster
Greedy
Rollout

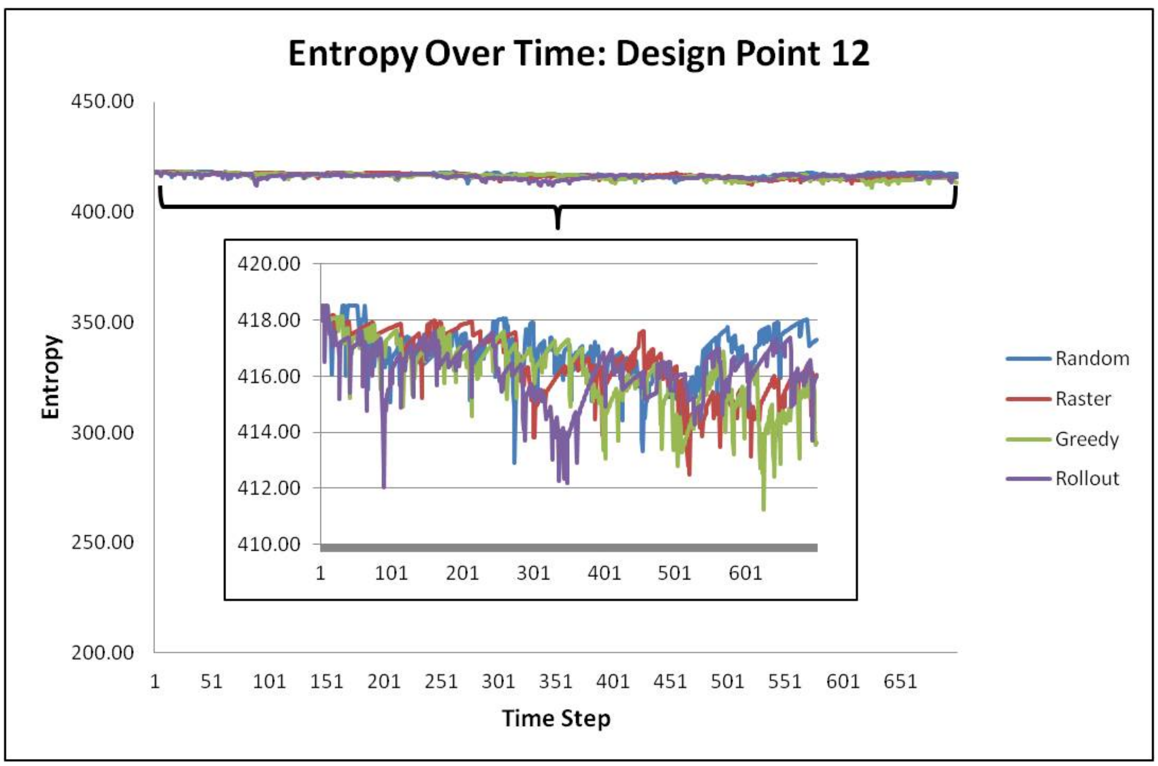


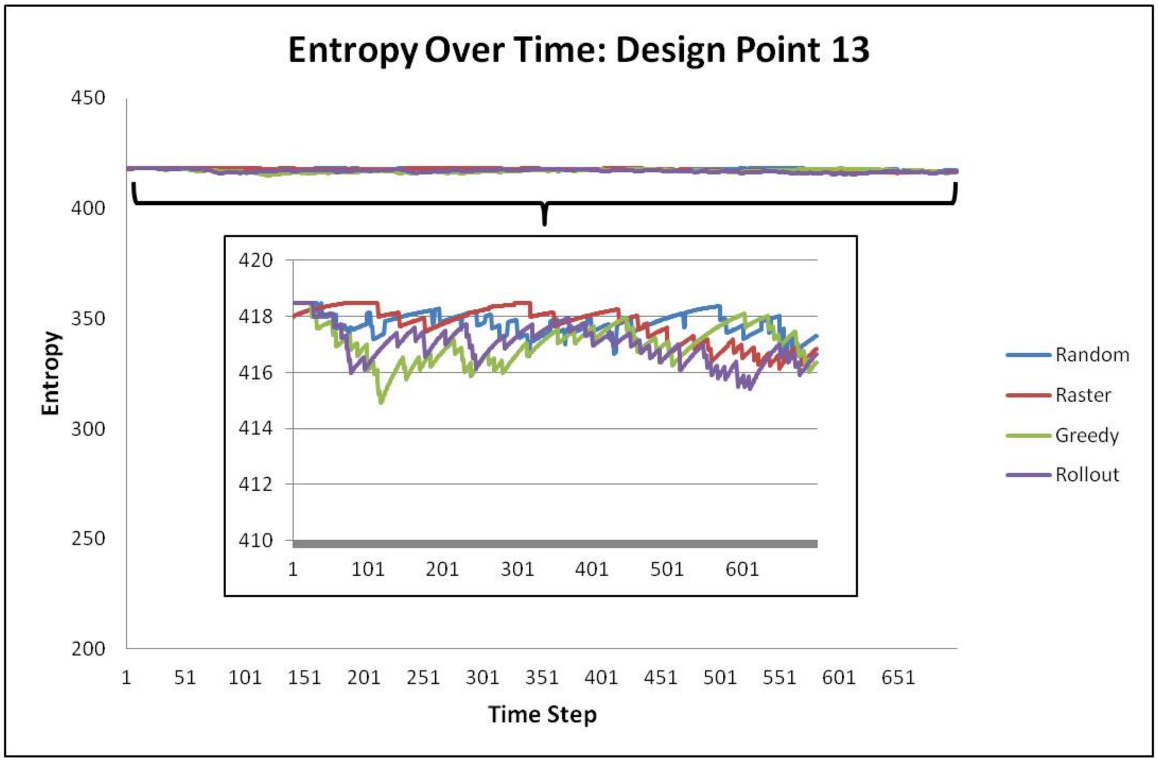
Entropy Over Time: Design Point 9
450.00
400.00
350.00
300.00
250.00
200.00
Entropy
1
51
101
151
201
251
301
351
401
451
501
551
601
651
Time Step
420.00
418.00
416.00
414.00
412.00
410.00
1
101
201
301
401
501
601
Random
Raster
Greedy
Rollout

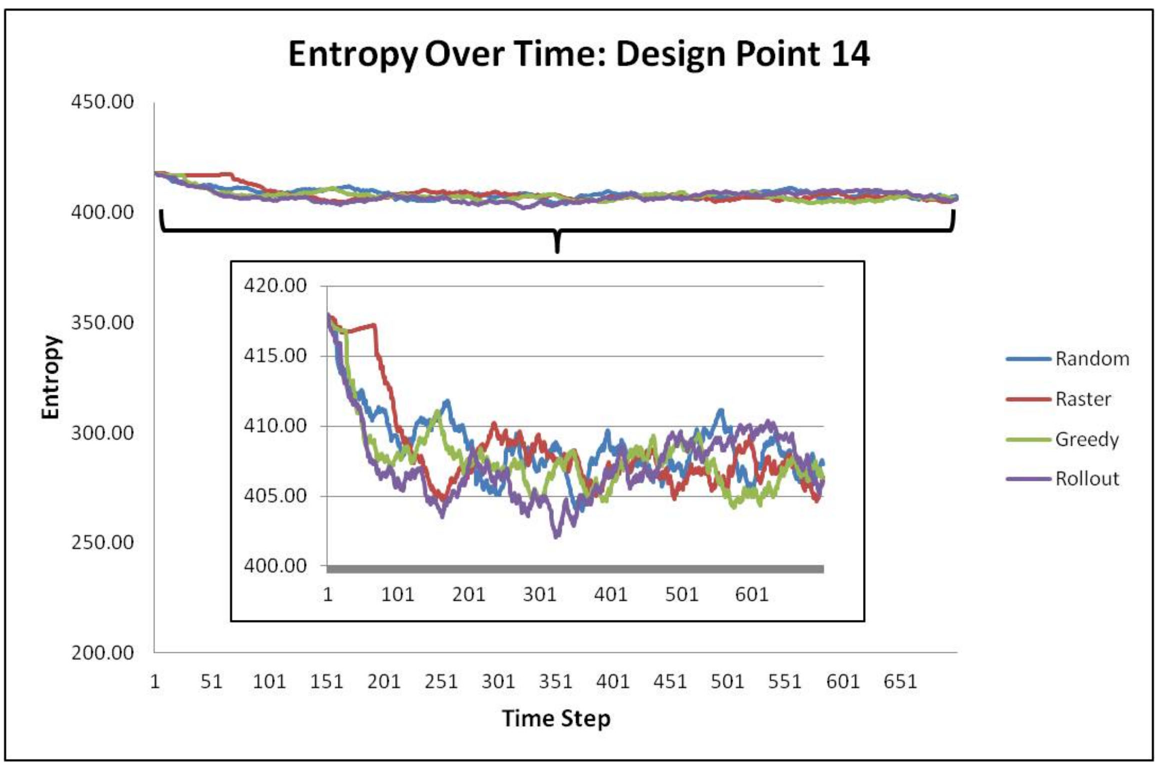


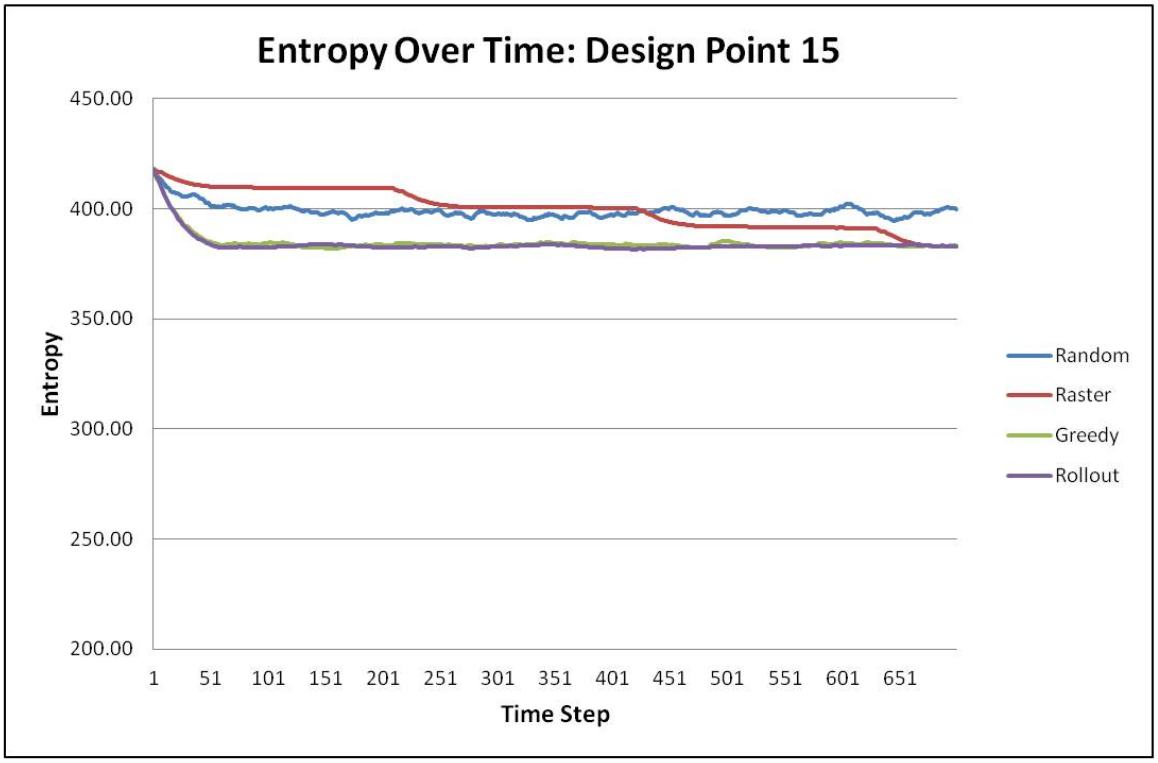
Entropy Over Time: Design Point 10
450.00
400.00
350.00
300.00
250.00
200.00
Entropy
1
51
101
151
201
251
301
351
401
451
501
551
601
651
Time Step
420.00
415.00
410.00
405.00
400.00
1
101
201
301
401
501
601
Random
Raster
Greedy
Rollout


Entropy Over Time: Design Point 11
450.00
400.00
350.00
300.00
250.00
200.00
Entropy
1
51
101
151
201
251
301
351
401
451
501
551
601
651
Time Step
Random
Raster
Greedy
Rollout

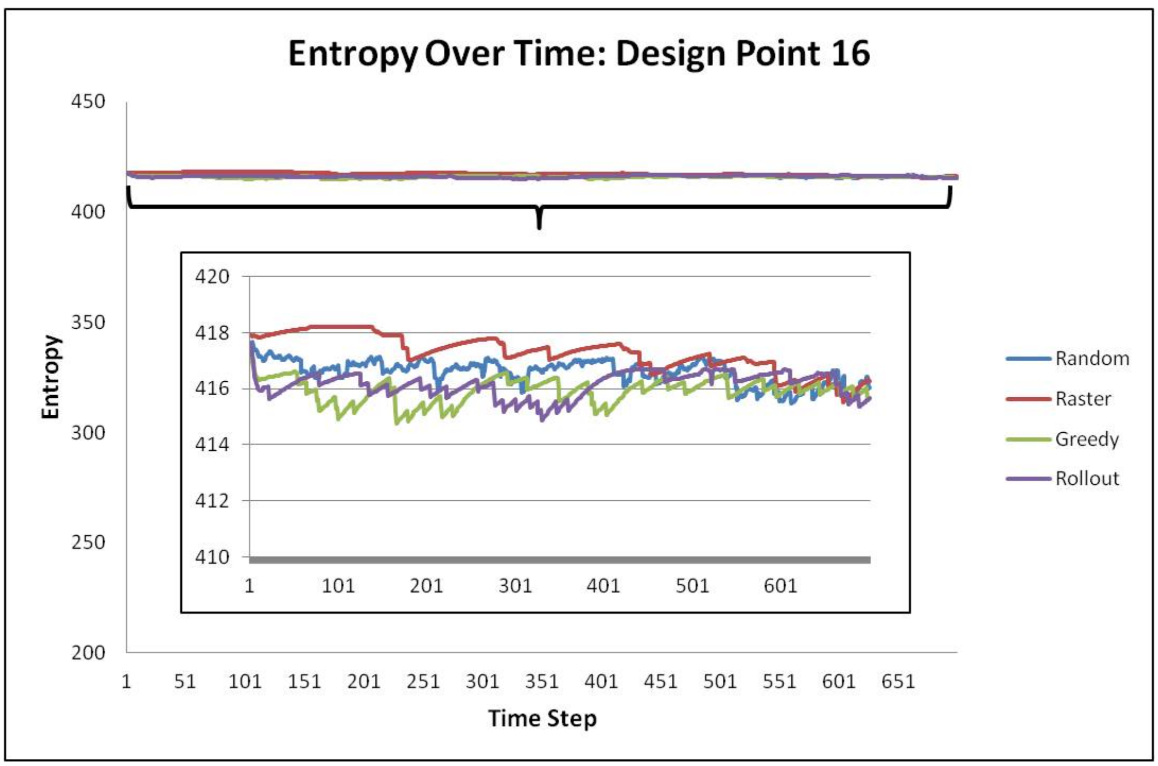

Entropy Over Time: Design Point 12
450.00
400.00
350.00
300.00
250.00
200.00
Entropy
420.00
418.00
416.00
414.00
412.00
410.00
1
101
201
301
401
501
601
1
51
101
151
201
251
301
351
401
451
501
551
601
651
Time Step
Random
Raster
Greedy
Rollout


Entropy Over Time: Design Point 13
Entropy
Time Step
Random
Raster
Greedy
Rollout

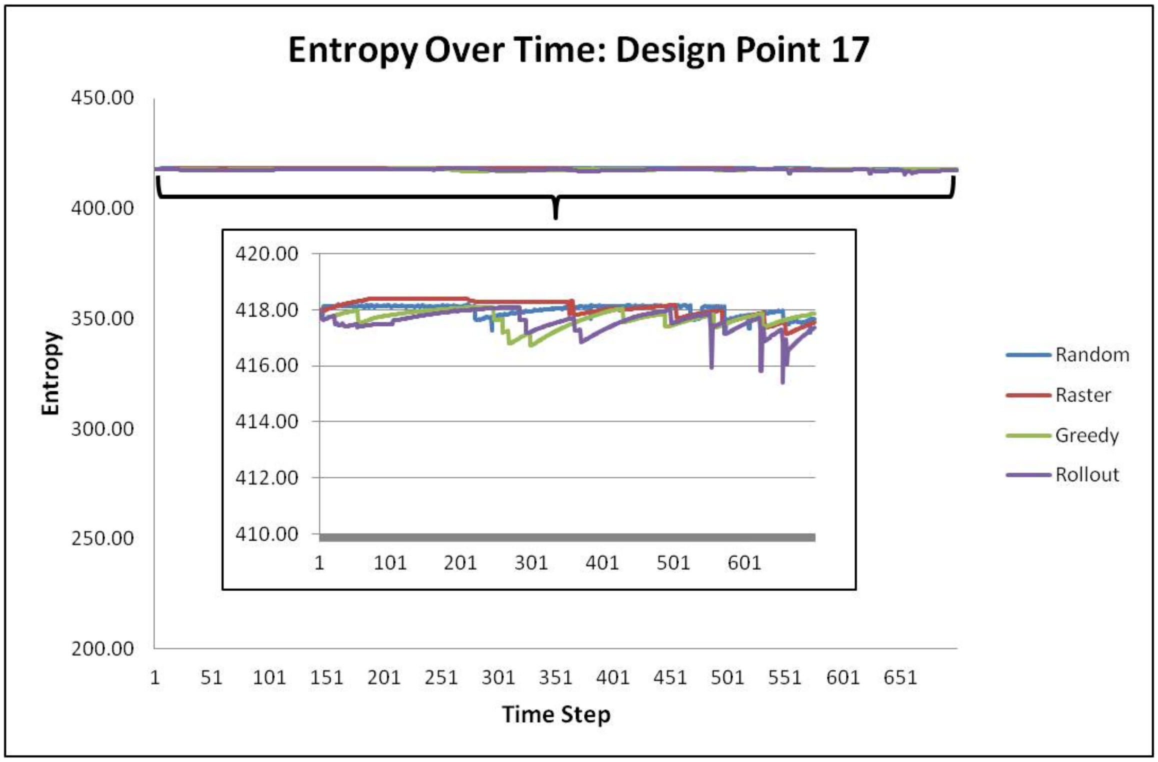

Entropy Over Time: Design Point 14
Entropy
Time Step
Random
Raster
Greedy
Rollout



Entropy Over Time: Design Point 15
Entropy
Time Step
450.00
400.00
350.00
300.00
250.00
200.00
1
51
101
151
201
251
301
351
401
451
501
551
601
651
Random
Raster
Greedy
Rollout



Entropy Over Time: Design Point 16
Entropy
Time Step
450
400
350
300
250
200
1
51
101
151
201
251
301
351
401
451
501
551
601
651
420
418
416
414
412
410
1
101
201
301
401
501
601
Random
Raster
Greedy
Rollout


Entropy Over Time: Design Point 17
450.00
400.00
350.00
300.00
250.00
200.00
Entropy
1
51
101
151
201
251
301
351
401
451
501
551
601
651
Time Step
420.00
418.00
416.00
414.00
412.00
410.00
1
101
201
301
401
501
601
Random
Raster
Greedy
Rollout

Surveillance versus Reconnaissance: An Entropy Based Model

Major Tamilyn S. Dismukes
Advisor: LTC Darryl K. Ahner
Department of Operational Sciences (ENS)
Air Force Institute of Technology

INTRODUCTION

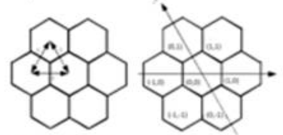
With the advancing capabilities of Intelligence, Surveillance, and Reconnaissance (ISR) assets and sensors, effective utilization of these resources continues to pose a challenge to military decision makers. There are numerous questions that a commander can ask in regards to this problem. What quantity and mix of ISR assets are needed to meet our national security challenges? How should ISR assets be used? What quantity and mix of surveillance and reconnaissance force is needed to cover a defined area? How do we measure our current ISR capabilities? As these questions are explored, two overarching measures of merit have been defined: discovery of new targets (reconnaissance) and persistence of already known targets (surveillance). Both are important to the situational awareness of the battle space, and tradeoffs between them should be explored and methodologies created that aid this exploration.

METHODOLOGY

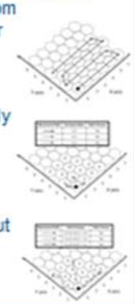
The methodology developed explored allocation of ISR assets while balancing detection of new targets versus surveillance of already detected targets using entropy as a measure of effectiveness. Scenarios with an unknown number of static and moving targets in a bounded geographical region were considered.

BASELINE MODEL

Hexagonal Coordinate System
- Equidistant Connectivity of Hexes
- Improved Diffusion Model



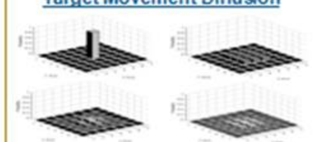
Search Algorithms
- Random
- Raster
- Greedy
- Rollout



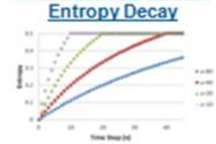
Model Inputs
- Grid and Sensor View Size
- # of Stationary/Moving Targets
- # of Search/Track Assets
- Sensor P_{Detect} and P_{Track}
- Number of Look Aheads (Rollout)

DIFFUSION MODEL AND ENTROPY DECAY

Target Movement Diffusion



Generalized Harmonic Entropy Decay



EXPERIMENTAL DESIGN

Nearly Orthogonal Latin Hypercube
- Space Filling Design
- 7 factors
- 17 design points

Two Scenarios
- Fixed Time Step
- Exhaustive Target Detection

Measures of Effectiveness
- Step Entropy
- Average Entropy
- Number of Targets Found
- Time Steps to Completion

RESULTS

Step Entropy
- Time series plots show the rollout algorithm decreased entropy during first 100 time steps and continues to remain as one of the lowest steady state entropy values throughout the runs

Average Entropy
- The rollout and greedy algorithms produced a significantly lower average entropy of the region then the raster and random algorithms

Number of Targets Found
- The rollout and greedy algorithms produced a significantly higher number of targets found in the exhaustive target detection scenario

Time Steps to Completion
- The rollout algorithm found all of the targets within the region in less time steps than the greedy and raster algorithm in the fixed time step scenario

CONCLUSIONS

The rollout search algorithm provides superior performance in the allocation of ISR assets while balancing detection of new targets versus surveillance of already detected targets.

Future Research
- Shift search algorithms within the run to possibly drive a lower entropy level
- Expand the scope of the model by including:
  - Heterogeneous Asset Force
  - Ground Truth and Different Diffusion Models
  - False Alarm Rate
  - Non-unique target identifiers
  - Targets of varying importance

AIR FORCE INSTITUTE of TECHNOLOGY

Bibliography

Ahner, D. (2009 Winter Simulation Conference). A Normalized Weighted Entropy Measure For Sensor Allocation Within Simulations., (p. 9). West Point, NY.

Barr, D. R., & Sherrill, E. T. (June, 1997). *Assessing Situational Awareness in Task Force XXI.* West Point, NY: Operations Research Center, USMA.

Barr, D., & Sherrill, E. T. (July, 1996). *Measuring Information Gain In Tactical Operations.* West Point, NY: Operations Research Center, USMA.

Beene, E. A. (1998). *Calculating A Value For Dominant Battlespace Awareness.* Wright Patterson Air Force Base, OH: AFIT.

Berry, P. E., & Fogg, D. A. (2003). *GAMBIT: Gueass-Markov and Bayesian Inference Technique for Information Uncertainty and Decision-making in Surveillance* Simulations. Edinburgh South Australia: Defense Science and Technology Organization.

Berry, P. E., Pontecorvo, C., & Fogg, D. A. (July, 2003). *Optimal Search, Location and Tracking of Surface Maritime Targets by a Constellation of Surveillance Satellites.* Edinburgh South Australia: DSTO Information Sciences Laboratory.

Bertsekas, D. P., & Castanon, D. A. (1999). Rollout Algorithms for Stochastic Scheduling Problems. *Journal of Heuristics* , 89-108.

Bertsekas, D. P., & Tsitsiklis, J. N. (1996). Neuro-Dynamic Programming.

Bertsekas, D. P., Tsitsiklis, J. N., & Wu, C. (1997). Rollout Algorithms for Combinatorial Optimization. *Journal of Heuristics 3* , 245-262.

Castanon, D. A., & Ahner, D. K. (2008). Team Task Allocation and Routing in Rosky Environments under Human Guidance. *47th IEEE Conference on Decision Control*, (pp. 1139-1144). Cancun, Mexico.

Cioppa, T. M., & Lucas, T. W. (July, 2007). Efficient Nearly Orthogonal and Space-Filling Experimental Designs for High-Dimensional Complex Models. *Technometrics* , Volume 49, Number 1, pp. 45-55.

Cormen, T. H., Leiserson, C. E., & Rivest, R. L. (1989). *Introduction to Algorithms.* Cambridge, MA: The Massachusettes Institute of Technology Press.

Hexagonal Coordinate Systems. (2005, February 13). Retrieved January 16, 2012, from The University of Edinburgh School of Informatics : http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/AV0405/MARTIN/Hex.pdf

Kleijnen, J. P., Sanchez, S. M., Lucas, T. W., & Cioppa, T. M. (2005). A User's Guide to the Brave New World of Designing Simulation Experiments. *INFORMS Journal of Computing* , Volume 17, Number 3, pp. 263–289.

Montgomery, D. C. (1997). *Design and Analysis of Experiments.* New York, NY: John Wiley & Sons, Inc.

Montgomery, D. C., & Runger, G. C. (1999). *Applied Statistics and Probability for Engineers.* New York, NY: John Wiley & Sons, Inc.

Murphy, E., & Payne, M. (2009). *Coverage, Control, and Reconnaissance: A New Paradigm for ISR Sufficiency Analysis.* Washington, DC: HQ USAF/A9IW.

Powell, W. B. (2007). *Approximate Dynamic Programming.* Hoboken, NJ: John Wiley & Sons, Inc.

Reza, F. M. (1961). *An Introduction to Information Theory.* New York, NY: McGraw-Hill Inc.

Sanchez, S. M. (2005). *Nearly Orthogonal Latin Hypercube Design Excel® Spreadsheet.* Retrieved February 12, 2012, from Naval Postgraduate SEED Center for Data Farming: http://harvest.nps.edu/

Shannon, C. E. (1949). *The Mathematical Theory of Communication.* Urbana, IL: University of Illinois Press.

Shupenus, J., & Barr, D. R. (1999). Information Loss Due to Target Mobility. *Military Operations Research* , 31-43.

Washburn, A. R. (2002). *Search and Detection, 4th ed.* Linthicum, MD: Institute for Operations Research and the Management Sciences.

Vita

Major Tamilyn S. Dismukes graduated from Portsmouth High School in Portsmouth, Rhode Island. She entered undergraduate studies at the United States Air Force Academy in Colorado Springs, Colorado where she graduated with a Bachelor of Science degree in Operations Research in May 2002 and received her commission as an Air Force Officer.

Her first assignment was at Hanscom AFB, Massachusetts, as a Space Warfare Research Scientist at the Air Force Research Lab Space Vehicles Directorate. From March 2005 to April 2006, she was the Tour Director, as well as a vocalist and dancer for Tops in Blue.

Her second assignment was at Edwards AFB, California, as the Assistant Flight Commander for the Tanker/Transport Electronic Warfare Test Flight and then as the Flight Commander for the Global Hawk Electronic Warfare Test Flight. While stationed at Edwards AFB, she deployed overseas in September 2007 to spend six months in Iraq as a Quality Assurance Representative with the Defense Contracting Management Agency. She also served as the Wing Executive Officer for the 412th Test Wing during that assignment.

In September 2010, she entered the Graduate School of Engineering and Management, Air Force Institute of Technology. Upon graduation, she will be assigned to HQ AF/A1 at Andrews AFB, MD.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. REPORT DATE *(DD-MM-YYYY)*
22-03-2012

2. REPORT TYPE
Master's Thesis

3. DATES COVERED *(From – To)*
Sep 2010 - Mar 2012

4. TITLE AND SUBTITLE
Surveillance versus Reconnaissance: An Entropy Based Model

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Tamilyn S. Dismukes, Major, USAF

5d. PROJECT NUMBER
N/A

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/ENV)
2950 Hobson Way, Building 640
WPAFB OH 45433-8865

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/OR-MS/ENS/12-09

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
"Intentionally Left Blank"

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution Statement A: Approved for Public Release; Distribution Unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
With the advancing capabilities of Intelligence, Surveillance, and Reconnaissance (ISR) assets and sensors, effective utilization of these resources continues to pose a challenge to military decision makers. The methodology developed explores allocation of ISR assets while balancing detection of new targets versus surveillance of already detected targets using entropy as a Measure of Effectiveness (MOE). Scenarios with an unknown number of static and moving targets in a bounded geographical region are considered. A baseline model was built to examine four different search algorithms: random, raster, greedy, and a rollout algorithm based on dynamic programming. A space-filling Nearly Orthogonal Latin Hypercube experimental design was applied to generate data to examine four MOEs: step entropy, average entropy, number of targets found, and time steps to completion. Based on statistical analysis and time series plots, the rollout algorithm's performance dominated others algorithms considered. In addition to minimizing uncertainty in the first 100 time steps of the run, the rollout algorithm also produced the highest number of targets found within the fixed time step scenario, and, for the exhaustive target detection scenario, discovered all of the targets within the region in less time steps. Based on these results, the rollout algorithm provides superior performance in the allocation of ISR assets while balancing detection of new targets versus surveillance of already detected targets.

15. SUBJECT TERMS
Intelligence, Surveillance, Reconnaissance (ISR), Entropy, Uncertainty, Rollout Algorithm

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
U	U	U	UU	79

19a. NAME OF RESPONSIBLE PERSON
Darryl K. Ahner, LTC, USA (ENS)

19b. TELEPHONE NUMBER *(Include area code)*
(937) 255-6565, ext 4708; e-mail: Darryl.Ahner@afit.edu